DSSI Installation and Maintenance Manual

Order Number EK–410AA–MG–001

February 1992

Revision/Update Information: This is the first draft of a new document. Please send comments to ESBTRN::DUNNE.

Digital Equipment Corporation
Maynard, Massachusetts

First Printing, February 1992

The Reader's Comments form at the end of this document requests your critical evaluation to assist in preparing future documentation.

FCC NOTICE: The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense may be required to take measures to correct the interference.

This document was prepared using VAX DOCUMENT, Version 1.2.

Contents

Chapter 3 Installation

Appendix A Cable Information

Appendix B Length Information

Appendix C VMS Naming Conventions for ISEs Connected to a KFQSA Controller

Glossary

Index

Figures

Tables

Preface

Purpose

This manual explains how to install and maintain DSSI VAXclusters.

Audience

This manual is written for the system administrator. A system administrator should be an experienced user who is familiar with VMS.

Related Documents

The following documents provide information related to DSSI VAXclusters:

- The *VMS VAXcluster Manual*
- The installation guides for VAX hardware products
- The VMS installation manuals for VAX products
- The *VMS Factory-Installed Software Installation Guide*
- The *VMS License Management Facility Manual*

Conventions Used

This manual uses the following conventions:

Convention	Meaning
Ctrl/*x*	While you hold down the Ctrl key, press another key or a pointing device button.
Ctrl/Alt/Del	While you hold down the Ctrl and Alt keys, press the Del key.
Return	Press the key that executes commands or terminates a sequence. This key is labeled Return, Enter, or ⏎, depending on your keyboard.
"enter"	Type all required text, spaces, and punctuation marks; then press Return, Enter, or ⏎, depending on your keyboard.
UPPERCASE	Uppercase letters in command syntax indicate commands and qualifiers. You can enter commands and qualifiers in any combination of uppercase or lowercase letters.

Convention	Meaning
lowercase	Lowercase letters in command syntax indicate parameters. You must substitute a word or value, unless the parameter is optional.
two-line commands	In VMS commands, a hyphen (-) at the end of a command line indicates that the command continues to the next line. If you type the hyphen and press Return, the system displays the _$ prompt at the beginning of the next line. Continue entering the command. If you do not type the hyphen, VMS automatically wraps text to the next line.
[ ]	Brackets in command descriptions enclose optional command qualifiers. Do not type the brackets when entering information enclosed in the brackets.
/	A forward slash in command descriptions indicates that a command qualifier follows.
. . .	A vertical ellipsis in an example indicates that not all the data is shown.
NOTE	Notes provide information of special importance.

Chapter 1

DSSI VAXcluster Basics: Hardware & Software

This chapter introduces the basic hardware and software elements of the Q–bus DSSI VAXcluster.

1.1 Introduction to VAXclusters

A VAXcluster is a highly integrated organization of VAX computers. As members of a VAXcluster system, computers can share processing resources, disks, and queues under a single VMS security and management domain and can boot and fail independently. Coupled with VMS Volume Shadowing, VAXclusters provide a high level of processing and data access.

In addition to the standard features of Local Area VAXclusters, DSSI VAXclusters offer a high availability of system resources, because several boot servers can access a common system disk and all data disks directly and can serve them to satellites. If one host fails, another host provides the satellites of the failed host with access to disks.

1.1.1 Types of VAXclusters

Clusters may be classified according to the hardware they use to access storage devices and to facilitate communication between individual nodes. This hardware is called an interconnect or bus. Interconnects differ in characteristics such as bandwidth, maximum usable distance, number of nodes supported, redundancy features, serviceability requirements, and cost. Five types of VAXclusters exist:

- Computer Interconnect (CI)
- Local Area VAXclusters (LAVcs), which use Ethernet as an interconnect
- Digital Storage Systems Interconnect (DSSI)
- Fiber Distributed Data Interconnect (FDDI)
- Mixed-Interconnect VAXclusters, which use more than one interconnect.

The CI and FDDI interconnects are not available on Q–bus VAX processors.

1.1.2 VAXcluster Software

The cluster software in all VAXcluster configurations is identical and includes the following components:

- **System Communication Services (SCS)** implements internode communication according to the Digital System Communication Architecture.

- **VAXport Drivers** (for example, PADRIVER, PIDRIVER, and PEDRIVER) interface cluster software to the hardware.
- The **Connection Manager** defines and coordinates the cluster. The Connection Manager uses SCS and provides an acknowledged message delivery service for higher VMS software layers. It also maintains cluster integrity when nodes join or leave the cluster.
- The **Distributed File System** allows processors to share mass storage, whether the storage disk or tape is connected directly to the processor or shared on common interconnects. All cluster-accessible disks can be made to appear as if they are local to every processor on the cluster. The distributed file system and VMS Record Managment Services (RMS) provide the same access to disks and files clusterwide as on standalone systems, to the record level.
- The **Distributed Lock Manager** synchronizes the functions of the distributed file system, job controller, device allocater, and other cluster facilities. The distributed lock manager provides a queuing function to put processes in a wait state until a particular resource is available.
- The **Distributed Job Controller** makes queues available clusterwide.
- The **Mass Storage Control Protocol (MSCP)** server and disk class drivers allow a processor to function as a storage controller. The server implements MSCP protocol and communicates either with a controller, for local disks, or directly with Digital Standard Architecture (DSA) disks, such as the RA series disks. Tape Mass Storage Control Protocol (TMSCP) serves the same function for tapes.

These components are always present on each cluster member so that if one member fails, the cluster continues to function. Users need only log in to another node to restart their processes.

1.1.3 VAXcluster Hardware

Cluster hardware has the following elements:

1. **VAX processors**; in this manual the VAX processors are Q–bus VAX or XMI-based VAX processors.
2. **Interconnects** or buses serve the functions of communication and storage for the processors. Currently two types of interconnects are available on Q–bus VAX processors: DSSI and Ethernet. The FDDI Interconnect is available indirectly through the use of a network bridge.
3. **Storage elements** provide storage using disks or tapes. The Hierarchical Storage Controller (HSC) is the storage device for CI clusters and the Integrated Storage Element (ISE) for DSSI VAXclusters. ISEs are compact storage elements that incorporate a traditional head disk assembly (HDA) or tape unit with an intelligent controller. (The term intelligent refers to complex functions, such as overhead and node communications, performed by the controller independently of the central processing unit (CPU).) The ISE is similar to an HSC with one

disk; however, the ISE controls one disk or tape only, whereas the HSC controls more than one.

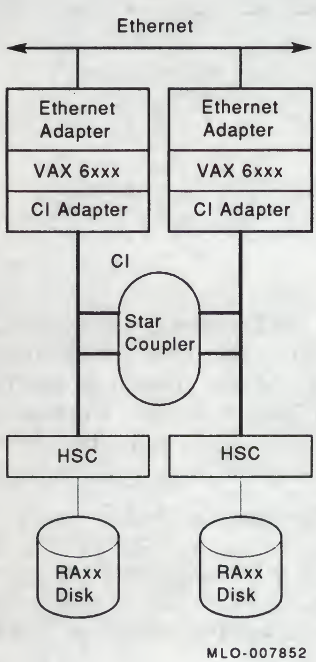
4. **Ethernet** performs network operations between processors within in all VAXclusters. In some clusters Ethernet also serves as the interconnect and supports both network and interprocessor (SCS) communications simultaneously.

5. **Port controllers** are intelligent controllers that connect the VAX processor to the interconnect. The port controller may be embedded on the CPU module or may exist as separate option that connects to the processor's Q–bus or to XMI.

1.2 CI VAXcluster

CI VAXclusters use the high-speed, dual-path CI interconnect to connect VAX processor nodes and hierarchical storage controller nodes. The Star Coupler is the common connection point for all cluster nodes. Cluster nodes may be any VAX processors specified in the VAXcluster Software Product Description (SPD), or they may be HSCs. A CI-only cluster may be converted to a mixed-interconnect configuration.

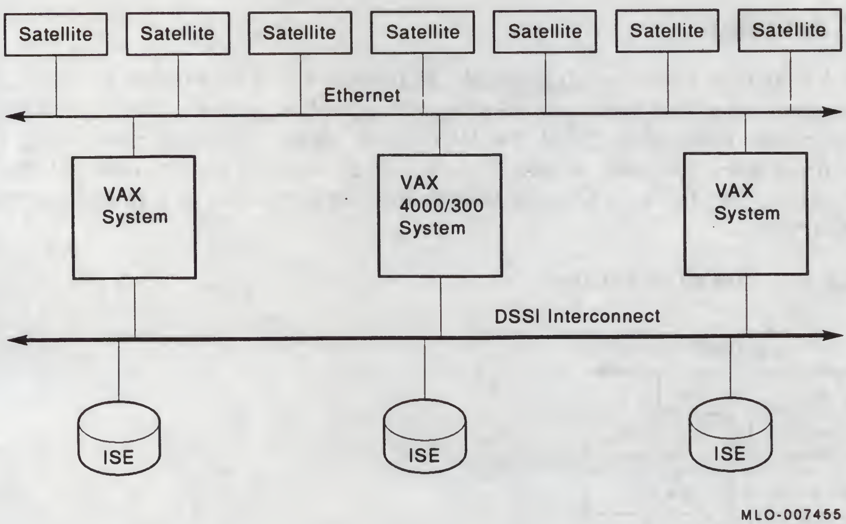
Figure 1–1: The CI VAXcluster



1.3 Local Area VAXcluster

On a Local Area VAXcluster interprocessor communication is carried out by means of an Ethernet port driver that emulates certain CI port functions. A cluster node may be any VAX or MicroVAX processor specified in the Software Product Description. A single Ethernet may support multiple Local Area VAXclusters. A single system may also use multiple LAN adapters in a single VAXcluster to carry interprocessor communication.

Figure 1–2: A Local Area VAXcluster

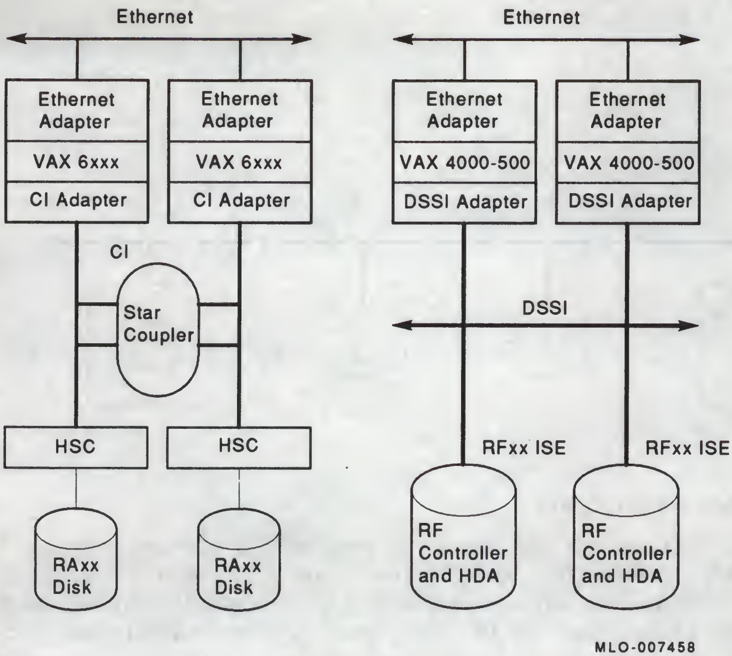


1.4 DSSI VAXcluster

A DSSI VAXcluster uses DSSI as the interconnect. VAX processors interface to DSSI by means of a DSSI adapter that is either embedded on the processor module or that exists on its own Q–bus or XMI module. As many as 8 *DSSI nodes* may use the same interconnect. A DSSI node is any device to which DSSI transports information and for which DSSI therefore needs an address, including ISEs and adapters on VAX systems. Figure 1–3 shows a DSSI VAXcluster.

To understand DSSI, consider a comparison between a cluster using the CI interconnect and a cluster using DSSI, as shown in Figure 1–3. The CI has two serial data paths, each with a peak bandwidth of 8.75 megabytes per second (70 megabits per second). It can reach 45 meters from the Star Coupler and can connect up to 32 nodes, with each node coupled by means of a transformer.

Figure 1–3: A Comparison of CI and DSSI VAXclusters



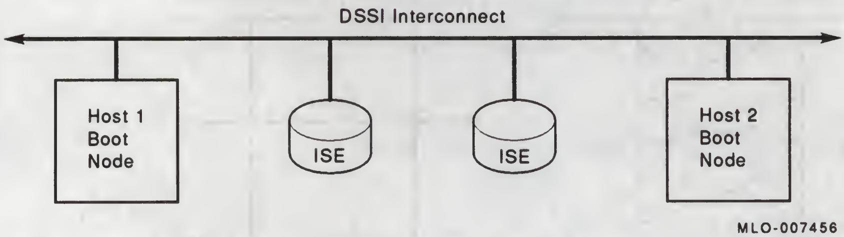
Because the DSSI interconnect is DC coupled, it does not include a serviceability feature that is available with CI: bus termination cannot be interrupted while the bus is active. Therefore, servicing of any adapter that contains termination resistors (for example, those used on the KA660 and KA640 VAX processor boards) can take place only after the cluster has been powered down.

1.4.1 Dual-Host DSSI VAXclusters

When DSSI was introduced, two nodes used it and shared RF-series ISEs. Ethernet carried the cluster traffic, so these configurations were called Dual-Host VAXclusters, but in actuality were Local Area VAXclusters with shared DSSI storage.

In Version 5.2, VMS provided support of interprocessor traffic on DSSI adapters embedded in the processors. The configurations with embedded adapters were called DSSI VAXclusters or if they used Ethernet to cluster satellites Mixed Interconnect VAXclusters. These two types of clusters were also referred to as Dual Host. New configurations allow DSSI VAXclusters to utilize three systems on a single DSSI interconnect.

Figure 1–4: Dual-Host VAXcluster



1.4.2 Three-System VAXclusters

Q–bus DSSI VAXclusters can be configured with processors ranging from the MicroVAX II to the VAX 4000 family. Introduction of the 4000-300 makes it possible to configure VAXclusters with three systems on single DSSI interconnect. Of the two embedded adapters on the 4000-300, one is not internally terminated, making it possible to add a third processor between those at either end of a single DSSI interconnect. This adapter has both an IN and an OUT connection and allows DSSI bus signals to travel through the adapter to another processor. The two connections also provide points of termination that could be utilized if the 4000-300 were configured as an end node on the interconnect.

1.5 Mixed-Interconnect VAXcluster

The Mixed-Interconnect VAXcluster originally used the CI interconnect and Ethernet. Now the term mixed-interconnect may refer to any combination of cluster interconnects.

1.6 Hardware Components of the Q–bus DSSI VAXcluster

The hardware components of the Q–bus DSSI VAXcluster are the DSSI interconnect, the DSSI adapters, and the integrated storage elements.

Figure 1–5: DSSI VAXcluster Using Three Host Systems

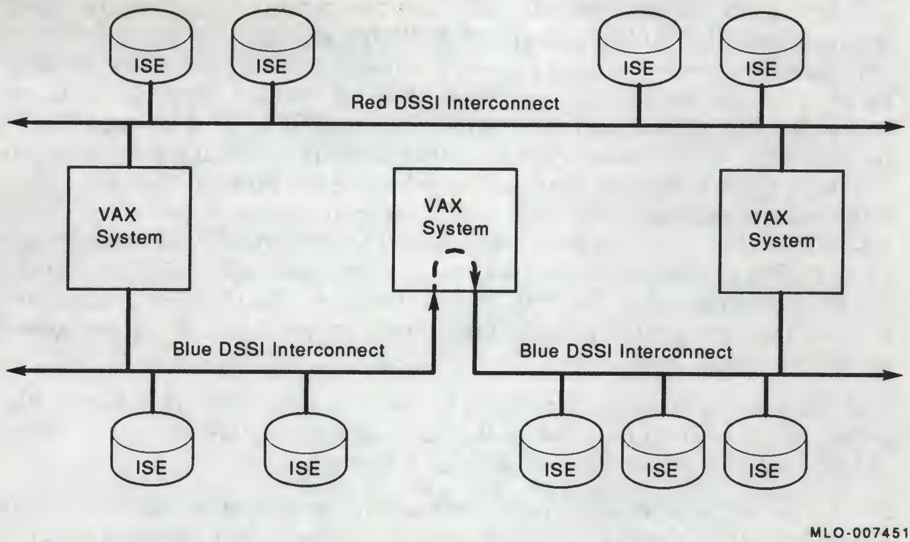
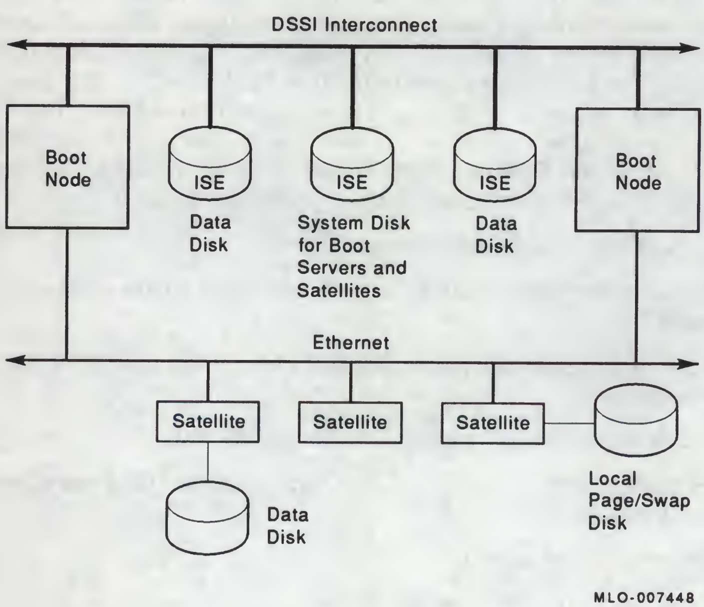


Figure 1–6: The Mixed-Interconnect VAXcluster



1.6.1 The DSSI Interconnect

The DSSI interconnect can be considered a micro CI interconnect. DSSI is physically different from CI, but logically the two are similar. In clusters using either interconnect, the multiple hosts of a VAXcluster system communicate directly with storage devices by using System Communication Architecture (SCA) protocols between host nodes and storage nodes. With CI, storage nodes are HSC-controller-and-disk combinations and with DSSI, ISEs. DSSI is an 8-bit parallel multidrop interconnect. When it was originally announced, its specified maximum distance was 6 meters. Since that time, time extensive testing has revealed that DSSI can provide reliable transmission of data for much longer distances. When three-system DSSI VAXcluster configurations were announced, DSSI distances were extended to support up to 82 feet (25 meters) within a computer room and up to 65.6 feet (20 meters) in the office environment. The difference between the two figures is accounted for by the fact that the ground differential typically is lower in a computer room than in an office environment.

The bus itself is actively terminated at each physical end. This termination must remain in place at all times while the bus is active. Digital does not support the use of DSSI without termination in place at both ends.

The actual physical implementation of the interconnect takes many forms. It can be flat-ribbon cable or round, sheathed cables (one type of round cable is used externally and another type internally), and in some of the newer enclosures (for example, the BA440) it can be implemented via an integral backplane.

Interconnecting DSSI options requires external round cables and bulkhead connectors, which connect the internal cables and the external cables. Two styles of DSSI connectors are used for these bulkhead interfaces: the P/S (Pedestal Style or Pin-Socket) style and the M/R (Midrange Style /Micro Ribbon) style. The appearance of these two connector styles is similar if not closely observed, but the internal connection components differ. Figure 2–1 illustrates the difference. PS style connectors are found on the Pedestal Style enclosures and systems. MR style are found on the 6000 series, SF arrays, and table-top TF85 systems.

Significant features of the interconnect are as follows:

- Single 8-bit parallel multidrop data path with both byte parity and packet error detection code (EDC)
- Peak bandwidth of 4 megabytes per second (of which more than 3.75 megabytes per second are usable)
- Maximum length (determined by qualified configurations)
- Up to 8 DSSI nodes: a maximum of 3 host system nodes (including adapters) and 5 storage elements.
- CMOS transceivers (DC-coupled)
- Low interconnect and node interface overhead
- Distributed, fair round-robin arbitration or fixed-priority arbitration

The DSSI interconnect is shorter. DSSI is also slower and less fault tolerant than the CI but it is much less expensive.

1.6.1.1 Termination

The first DSSI systems introduced DSSI adapters that provided integral termination of the bus. It was not possible to locate these adapters anywhere except at the ends of the bus. Implementations of newer adapters (such as the SHAC and KFMSA-BA) do not automatically embed termination within themselves. Instead, termination is provided by means of a separate terminator, which can be plugged into a standard DSSI bulkhead connector. It is possible to locate the new adapters at locations on the bus other than at the ends. These adapters make it possible to configure DSSI VAXclusters with three systems on the same DSSI interconnect.

1.6.1.2 Serviceability

The design of DSSI and VMS software makes it possible to remove and replace Intelligent Storage Elements (ISEs) on a DSSI bus while it continues to perform normal operations with the remaining nodes. This feature makes it possible to repair failed ISEs while the remaining parts of a DSSI VAXcluster continue to provide service to users. It is now possible to perform online service on DSSI adapters that do not include embedded termination.

A major difference between CI and DSSI is that all devices on CI, which is AC coupled, can be serviced without powering down the interconnect, whereas some devices on DSSI, which is DC coupled, cannot. With DSSI, servicing of any adapter that contains termination resistors (for example, those used on the KA660 and KA640 VAX processor boards) can take place only after the cluster has been powered down. In addition to adapters that do not include embedded termination, devices that can be serviced while the interconnect is running are ISEs in supported enclosures and SF72, SF73, TF857 storage enclosures. For a list of supported enclosures, see Section 1.6.4.

1.6.2 Integrated Storage Elements

RF-series and TF-series ISEs are part of the continuing evolution of Digital Storage Architecture. RF-series ISEs are disks, while TF-series ISEs are tape. Like the HSC, each ISE contains an MSCP server and communicates with host adapters through DSSI in the same way that HSCs communicate through the CI to their host adapters. Both use the protocols defined by Digital's System Communication Architecture (SCA).

An ISE is analogous to an HSC with one RA disk connected; the DSSI is analogous to the CI; and the embedded DSSI adapter on the VAX 4000-500 CPU board (KA680) is analogous to an XCD on a VAX 6000.

DSSI supports up to seven ISEs daisy-chained through a single cable to an adapter in the host. DSSI adapters can be embedded within a CPU module, such as the KA680, KA670, and KA660, or can be separate (such as the KFQSA) and plug into the Q–bus.

1.6.3 DSSI Adapters

Currently four DSSI adapters are available: the SHAC, the embedded single-host adapter chip used on the KA660, KA670, and KA680 processor boards; the EDA640, the KA640 embedded adapter; the KFQSA Q–bus to DSSI protocol converter; and the KFMSA, the XMI-DSSI adapter. The following table describes the adapters.

Table 1–1: Adapters and Processors

Adapter	Systems	CPU Board	Maximum Adapters per CPU
KFMSA	VAX 6000s and 9000s	–	Implements 2 DSSI buses. This is the XMI to DSSI adapter.
SHAC	VAX 4000-500	KA680	2 interfaces embedded on the processor board (one with IN/OUT connectors) and 2 KFQSAs on Q–bus
SHAC	VAX 4000-300	KA670	2 interfaces embedded on the processor board (one with IN/OUT connectors) and 2 KFQSAs on Q–bus
SHAC	VAX 4000-200	KA660	1 interface embedded on the processor board and 2 KQFSAs on Q–bus
EDA640	MicroVAX 3300/3400	KA640	1 EDA640 embedded on the processor board and 2 KQFSAs on Q–bus
KFQSA	All Q–bus VAX and MicroVAX systems		This is the Q–bus to DSSI adapter.

Single-Host Adapter Chip (SHAC)

The single-host adapter chip (SHAC) is a single-chip, Very Large Scale Integration (VLSI) version of Digital's systems communications architecture (SCA) port that uses a DSSI interconnect. The SHAC functions as the embedded adapter on the VAX 4000 processors as well as well as on the KFMSA, the XMI to DSSI adapter. The SHAC is implemented as a RISC architecture CPU with processor performance rated at 10 MIPS.

CI, another SCA realization, has defined a port driver/port interface that has been used to connect VAX systems in clusters. DSSI has adopted the same interface; therefore, the same port driver can serve either a CI port or a SHAC port. The SHAC can be used to connect a host to any other device that can communicate through the CI-DSSI protocol. The SHAC provides a high-performance method of communicating with disk and tape ISEs as well as supporting cluster communication with other embedded adapters and with the KFMSA unembedded adapter. The SHAC is the embedded adapter used on the VAX 4000-500 (two are used), 4000-300, and 4000-200 CPUs and also on the KMFSA (XMI-to-DSSI) adapter.

EDA640

The EDA640, an embedded DSSI adapter, is implemented on the KA640 processor board by the SII chip, the DXX chip, and four 32K–by–8-bit static RAMs. The SII and DXX can transmit or receive packets between the static RAM of the KA640 module and another DSSI node without CPU intervention; however, *higher CI port*

functionality requires software assistance of the driver. The MSCP, TMSCP, and DUP class drivers interface to the EDA640 through PIDRIVER, an SII port driver in VMS. Like the SHAC, the EDA640 supports both host-host and host-ISE communication.

The KFQSA Q–bus–to–DSSI Adapter

The KFQSA is a quad-height Q–bus system module. It emulates a Storage System Port (SSP) controller, such as an RQDX3, KDA50 or TK70, for each ISE connected to the KFQSA. However, the disk or tape controller is actually located in the ISE, and the KFQSA simply provides the communication mechanism that allows the host to communicate with the ISE. Because of its architectural design, the KFQSA does not support host-to-host communication. In configurations using KFQSAs, cluster communication is handled by Ethernet, using the same driver that supports Local Area VAXclusters, and DSSI handles only storage.

KFMSA XMI–DSSI Adapter

The KFMSA is the adapter used in the VAX 6000 series of processors. It is SCA compliant and implements two DSSI buses on one SMI option card. The KFMSA-BA variant, which may utilize a bulkhead cabinet kit with IN/OUT connectors, can be terminated on either the adapter board or on the bulkhead. This enables the configuration of three-system DSSI VAXclusters and also provides for online serviceability of the option. To implement termination on the bulkhead, you need the cabinet kit with the order number CK-KFMSA-LN.

1.6.4 DSSI Expansion Enclosures

The following expansion enclosures enhance the storage capacity and allow for greater flexibility in configurations:

- **The B400X** provides compartments for 3 ISes and 12 Q–bus modules.
- **The B213F** provides compartments for 2 ISEs and 12 Q–bus modules.
- **The R23F** provides compartments for 2 removable ISEs.
- **The R215F** provides compartments for 3 ISEs.
- **The R400X** provides compartments for 7 ISEs and two separate DSSI interconnects.
- **The SF72** provides compartments for 2 or 4 (1GB) RF72 ISEs.
- **The SF73** provides compartments for 2 or 4 (2GB) RF73 ISEs.
- **The SF100** provides a TF857 magazine tape subsystem and space for an SF72 or SF73 storage building block. These together hold 5 ISEs.
- **The SF200** provides compartments for 6 SF72 or SF73 storage building blocks, each of which holds 4 ISEs, and 2 TF857s, each of which holds 1 ISE.
- **The SF210** provides compartments for 24 ISEs (6 SF73 storage arrays, each of which holds 4 ISEs).
- **The TF857** provides a compartment for 1 TF ISE.

1.6.5 Location of DSSI Connectors on Hardware Components

The following pages show the locations of DSSI connectors on DSSI hardware components.

Figure 1–7: DSSI Connectors on a VAX 4000-500 or 4000-300

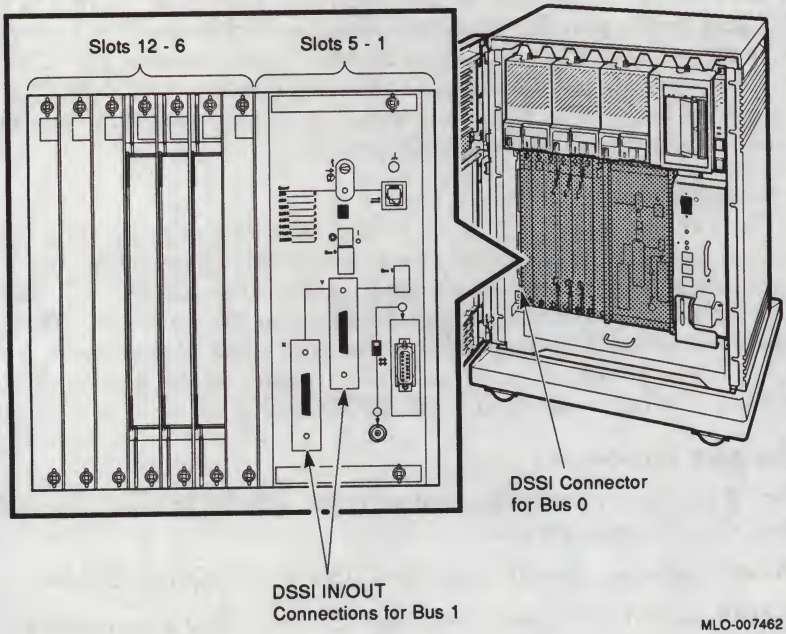
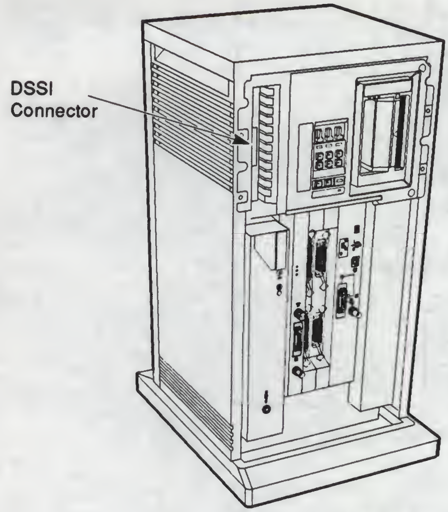
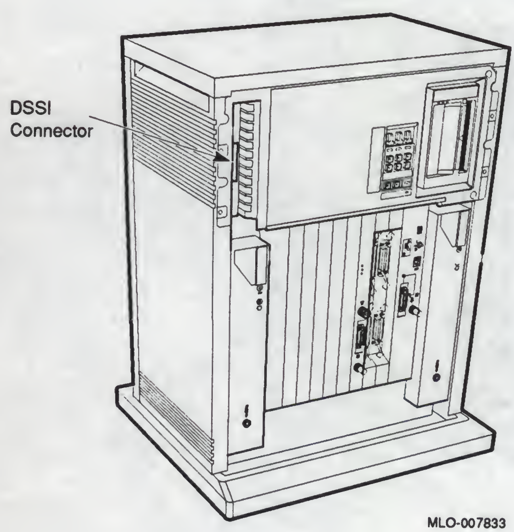


Figure 1–8: DSSI Connectors on a BA215F Expansion Enclosure



MLO-007461

Figure 1–9: DSSI Connectors on a BA213 Expansion Enclosure



MLO-007833

Figure 1–10: DSSI Connectors on a B400X Expansion Enclosure

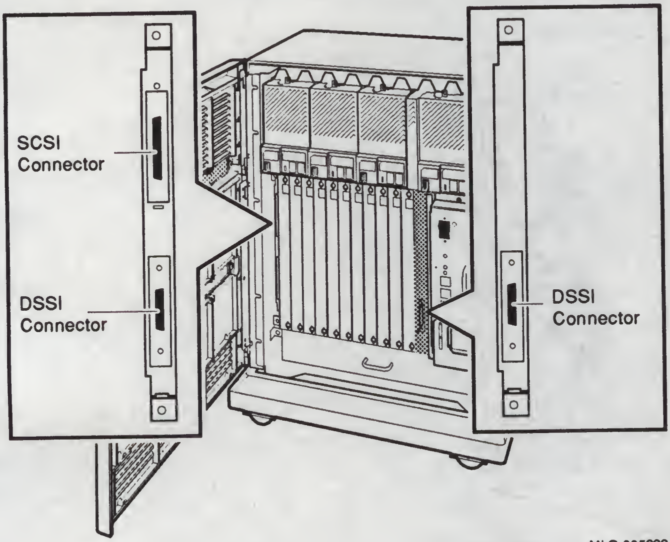


Figure 1–11: DSSI Connectors on an R400X Expansion Enclosure

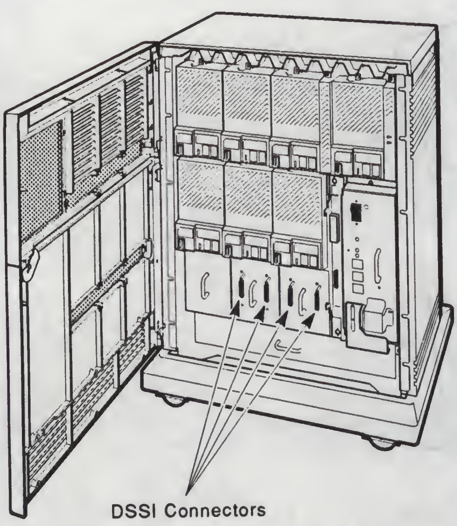


Figure 1–12: DSSI Connectors on an R215F Expansion Enclosure

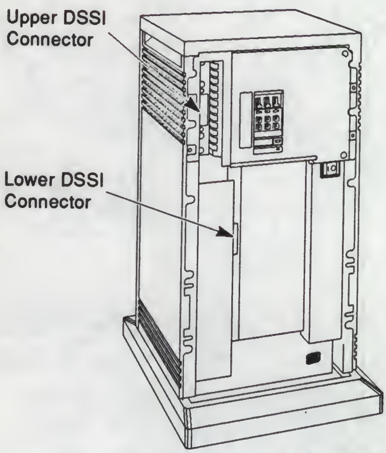


Figure 1–13: DSSI Connectors on a TF857 Storage Enclosure

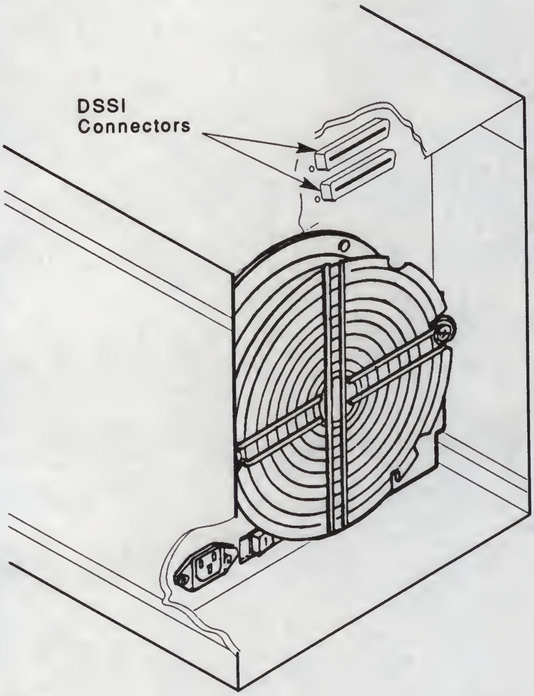


Figure 1–14: DSSI Connectors on an SF72/SF73 Storage Building Block

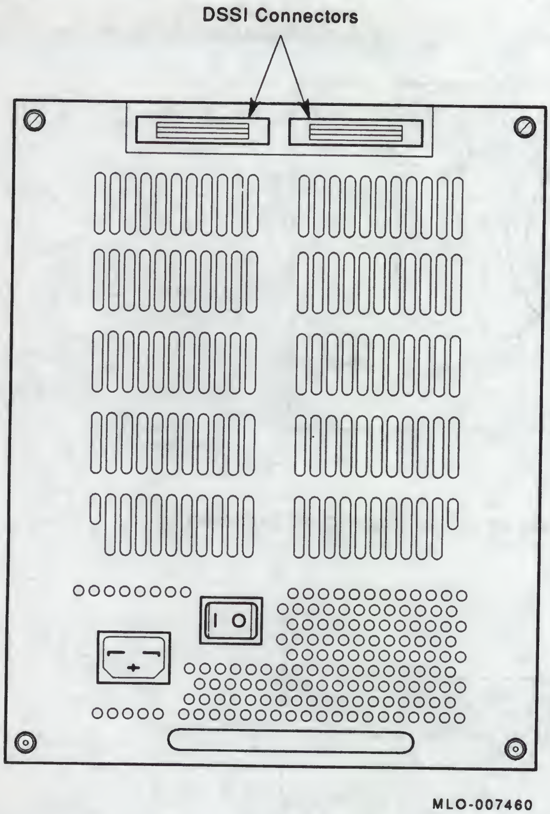


Figure 1–15: DSSI Connectors on a TF85 Table-Top Storage Enclosure

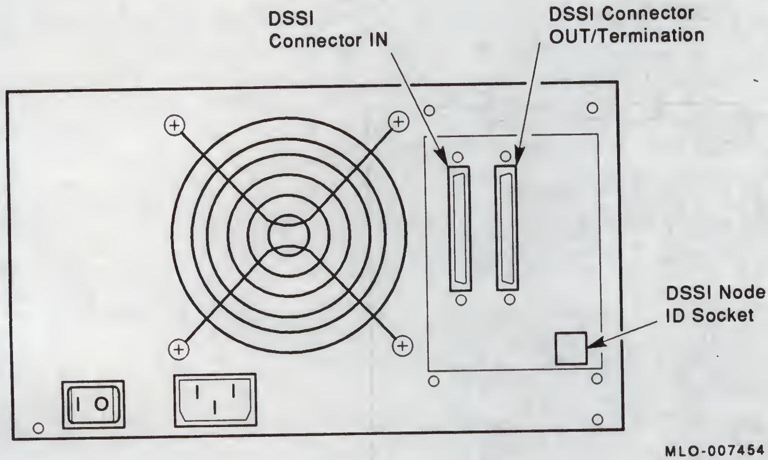
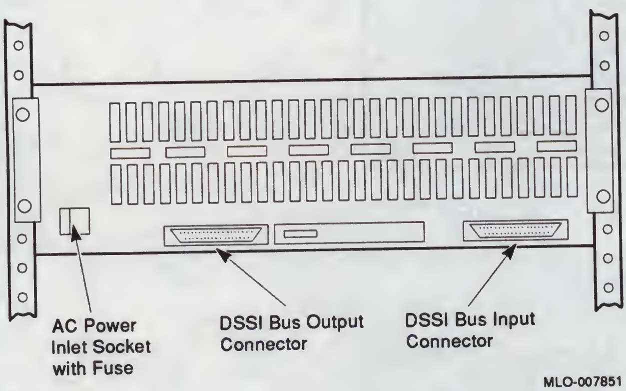


Figure 1–16: DSSI Connectors on an RF23F Expansion Enclosure



Chapter 2
Configuration Rules and Recommendations

This chapter describes rules that must be implemented in order to successfully configure and maintain a DSSI VAXcluster. It also includes recommendations that are intended to help you in configuring a VAXcluster that meets the particular needs of your users.

2.1 Configuration Rules for All VAXclusters

The following rules apply to all VAXcluster configurations:

- The maximum number of VAX systems supported in a VAXcluster system is 96. The maximum number of VAX systems supported by a single DSSI interconnect is 3.
- The following systems are not supported in any VAXcluster:

  VAXstation I
  MicroVAX I
  VAX–11/725
  VAX–11/730
  VAX–11/782

- All VAX 9000, 6000, 85xx, 86xx, 8700, and 88xx series VAX systems that participate in a VAXcluster system must be connected by CI, DSSI, or FDDI. An Ethernet VAXcluster system may include a maximum of one of these VAX systems.
- A VAX system or storage controller may not participate in more than one VAXcluster system at a time.
- The Rule of Total Connectivity must be met. This states that in a VAXcluster system every VAX system must be able to communicate directly with every other VAX system. VAXcluster nodes do not perform routing for interprocessor messages.

2.2 Configuration Rules for Q–bus DSSI VAXclusters

The configuration rules listed in this section are based on hundreds of hours of extensive qualification testing of various VAXcluster configurations. These system tests were performed to ensure the proper operation of all VAXcluster components as they interact in the various configurations that are possible. Of particular importance are the limits set for individual DSSI bus lengths used within a configuration. Violation of these rules will produce configurations that Digital may not support.

Configuration rules may change in the future to incorporate configuration paradigms not currently supported. Please check with a Digital representative before attempting any configuration not covered by these rules.

Rules for DSSI VAXclusters are based on the signal-integrity capacity and system-level compatibility of DSSI interconnects. A major concern in configuring a DSSI VAXcluster is to ensure that the maximum length of each individual DSSI interconnect (measured from end terminator to end terminator and including all interconnecting cables and internal enclosure wiring) does not exceed a maximum qualified length. Currently these lengths are 82 feet (25 meters) in a computer room and 65.6 feet (20 meters) in an office environment.

2.2.1 Five-Enclosure Rule

To simplify the configuration process with Q–bus DSSI VAXclusters, a quick configuration rule has been established with certain restrictions: up to 5 enclosures can be configured on each DSSI bus; for example, two VAX systems and up to three expansion enclosures or three VAX systems and up to two expansion enclosures. If you adhere to the restrictions listed here you can quickly and easily determine if a particular Q–bus configuration is supported. This rule eliminates the need to perform length calculations. For a list of valid enclosures, see Table 2–1. Note that this rule can be applied only with the use of the following cables: BC21M-09 (a 9-foot cable that connects Q–bus VAX hardware) and BC22Q-09 (a 9-foot cable that connects Q–bus VAX hardware and the SF72, SF200, TF85, TF857), and BC21Q-3F (a 40-inch internal cable that can be used to connect a TF857 to an SF7x, and SF7xs to each other in the same rack). The following restrictions apply to the 5-enclosure rule:

- The configuration cannot include VAX 6000 processors or SF2x0 storage arrays.
- Only one SF100 per DSSI bus is supported (the TF857 and optional SF7x storage building block each count as one enclosure).
- You must use the standard 9-foot interconnect cables (BC21M-09, BC21Q-09, and BC22Q-09) for intercabinet connections.

2.2.2 Additional Configuration Rules

If you cannot use the 5-enclosure rule, it is necessary to calculate the entire length of each DSSI bus you configure by adding up the lengths of interconnect cables as well as any DSSI bus segments that may be present in each of the enclosures you may be cabling. See Appendix A for a table that lists the DSSI components and their lengths. In the computer room, each bus cannot exceed 82 feet (25 meters). In the office environment, it cannot exceed 65.6 feet (20 meters). The lower length in the office environment is due to the poorer noise margins characteristic of office wiring and its typical ground differentials.

The enclosures listed in Table 2–1 are valid for DSSI VAXclusters.

Table 2–1: Q–bus Enclosure Characteristics

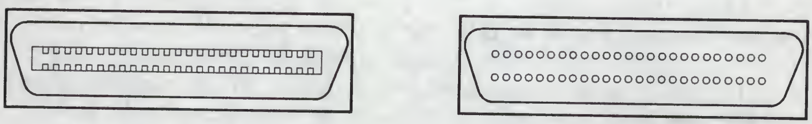
Enclosure	No. of Disks	Cable Connector Type
BA440 VAX 4000-300 or higher	4	/PS (Pedestal style)
BA430 VAX 4000-200 Cab.	4	/PS
R400X	7	/PS
B400X	4	/PS
SF100[1]–SF72/73	2 or 4 RF72/73s	/MR (Midrange style)
SF100–TF857	1 (tape only)	/MR
TF85 (Table-Top)	1	/MR
B213/213F	3	/PS
BA215	2	/PS
R215F	3	/PS
R23F	2 (CSS removable disk packs only)	/PS

[1]Only one SF100 per interconnect is supported.

NOTE

Two different types of connectors are used with DSSI. It is important to be aware of this, so that you do not accidentally damage your equipment. The two types of connectors are shown in Figure 2–1.

Figure 2–1: Two Types of Connectors Used in DSSI



The following rules apply to all Q–bus DSSI VAXcluster configurations:

- All systems must be Q–bus VAX or MicroVAX systems.

- A minimum of two MicroVAX, VAX 4000 and a maximum of three must be present.
- All VAX systems connected to the same DSSI interconnect must be members of the same VAXcluster.
- The VAX systems must have VMS and VAXcluster licenses.
- One VAX system must have a DECnet full-function license; the other systems can have DECnet end-node licenses. DECnet communications are not supported by DSSI but are required for VAXcluster operation.
- Each system must have Ethernet network hardware.
- Each DSSI interconnect must be terminated at both ends at all times, either by an adapter or by a DSSI terminator.
- More than one DSSI adapter type is supported on VAX systems. See Table 1–1 for the maximum number of DSSI adapters that can be utilized per system.
- The number of DSSI nodes (including VAX systems and ISEs, each of which count as one DSSI node) cannot exceed 8.
- Each node on a single DSSI interconnect must have a unique DSSI ID number, which allows the software to communicate with the storage devices. Numbers must be between 0 and 7, because these numbers are permanently associated with the hardware. Node numbers may be repeated on different DSSI interconnects that are connected to the same VAX systems.
- Between connectors, the maximum single cable length is 25 feet (7.6 meters).
- A common ground must exist between all elements (VAX systems and storage elements), and a ground strap must be used between cabinets. The ground offset must be less than 200 mv. peak to peak.

2.3 Configuration Recommendations for DSSI VAXcluster Systems

The configuration rules listed in the previous section provide the minimum guidelines necessary to follow in creating a supportable VAXcluster configuration. These rules do not guarantee the optimum or best configuration for any particular application. In, fact it is quite possible to configure a VAXcluster that obeys all the rules listed but yet does not perform optimally, or even adequately, for the tasks it must perform.

Configuring a VAXcluster to meet specific needs must take into account many factors in addition to the basic rules of supportability. Consideration must be given to the compute capacity required, I/O capacity needed, and the degree of data availability preferred (including such questions as how and where to locate hardware; how to use volume shadowing and error recovery; and how to plan for servicing needs).

Many general VAXcluster configuration recommendations apply to both DSSI and CI VAXclusters. The manual entitled *Guidelines for VAXcluster System Configurations*

provides a discussion of general VAXcluster principles. This section deals with recommendations for DSSI VAXcluster configurations only.

2.3.1 Assigning DSSI Node ID Numbers

Highest DSSI node ID numbers (7, 6, and 5) should be used for host nodes. Lower numbers should be used for RF and TF ISEs. Tape ISEs typically have the lowest numbers, often 0.

You set the DSSI node ID numbers by inserting the plugs that come with your VAX systems in the appropriate location in the processor and ISE. Consult the documentation accompanying your hardware for information on this location.

2.3.2 Adapter Differences

Differences among adapters make it necessary to consider adapter performance when allocating resources. Currently four host adapters provide access to the DSSI bus, as follows:

- EDA640: the embedded adapter used on the KA640
- SHAC (Single Host Adapter Chip): the embedded adapter used on the KA660, KA670, KA680
- KFQSA: the Q–bus-to-DSSI adapter used on Q–bus VAX machines
- KFMSA: the XMI-to-DSSI adapter used on VAX 6000s

The SHAC adapter services between 800 and 1200 I/O requests per second, as opposed to 170 per second for the KFQSA. The EDA640 and the SHAC also carry interprocessor traffic, while the KFQSA does not, making it necessary to use Ethernet for this purpose. Ethernet is still required to carry network traffic in all VAXclusters and to carry cluster communications for booting satellities. Another factor affecting the KFQSA is that the Q–bus contains addresses for ISEs. Therefore, adding a KFQSA adapter, or programming additional ISEs on it, may require removing options and changing addresses. These and other differences among adapters are summarized in Table 2–2.

Table 2–2: Adapter Differences

Adapters	Cluster Traffic Support	Multihost Support	I/Os per second	Type	Online Serviceability
KFMSA	Yes	Yes	800 x 2	XMI	Yes, BA variant; No, AA variant
SHAC (KA680)	Yes	Bus 0-No Bus 1-Yes	1200 x 2	Embedded	
SHAC (KA670)	Yes	Bus 0-No Bus 1-Yes	800 x 2	Embedded	
SHAC (KA660)	Yes	No	800	Embedded	No

Table 2–2 (Cont.): Adapter Differences

Adapters	Cluster Traffic Support	Multihost Support	I/Os per second	Type	Online Serviceability
EDA640	Yes	No	340	Embedded	No
KFQSA	No	No	170	Q-Bus	No

For high performance, ensure that like VAX systems and like adapters are connected together whenever possible.

2.4 Number and Type of Processors Supported

At the time this manual is being written, DSSI VAXclusters can be configured with one, two, or three processors ranging from the MicroVAX II to the VAX 4000 family. Introduction of the 4000-300 makes three-system DSSI VAXclusters possible. One of the two embedded adapters on the VAX 4000-300 (and 4000-500) is unterminated, making it possible to attach the third processor between those on either end of the interconnect. The unterminated adapter has both an IN connection and an OUT connection, which allow signals to travel through the adapter to another processor. This SHAC adapter is terminated on the processor-handle DSSI bulkhead. Dual-host systems prior to the introduction of the VAX 4000-300 required the processors to be positioned at the ends of the interconnect.

2.5 Number of Interconnects, Adapters, and ISEs Supported

The number of DSSI buses any processor can support is a function of the number of embedded adapters that reside on the processor module as well as the number of DSSI adapters that may be installed in the Q–bus. Today the number of KFQSAs (Q–bus to DSSI adapters) that may be installed is two, although certain configurations may be qualified with a higher number. At present, the Q–bus can support up to 2 storage adapters (including the KFQSA, RQDX3, KDA50, KZQSA) of any type in any combination.

One VAX 4000-300 or higher processor is required for each bus to which all three processors attach. In the Q–bus space only, the SHAC port 1 adapter on the VAX 4000-300 (or higher) is the unterminated adapter.

2.6 Performance Considerations

For optimum VAXcluster performance, consideration should be given to performance differences among the processors. Although DSSI supports processors of many performance levels, clustering processors of widely disparate performance levels may have undesirable consequences. The disparity may be apparent to users and cause the workload on processors to become unbalanced. This is especially true where users are connected to systems via local area transports (LATs) and therefore have a choice as to which system they use. Performance disparities also make capacity planning under failure conditions difficult. A further negative consequence

of unbalanced performance levels among processors is that adapter performance also varies and may increase the disparity.

2.6.1 Configuring for General High Performance

Where performance levels are different, decisions regarding where applications are run and the capacity level remaining in a cluster when VAX systems are removed are significant. For example, a VAX 4000-300 and two MicroVAX IIs is not a good combination, because the 4000-300 is more than eight times more powerful than a MicroVAX II. Removal of the 4000-300 in such a cluster represents an 80 percent loss of capacity.

For the reasons listed in Section 2.6, Digital strongly recommends that older MicroVAX II and MicroVAX 3xxx systems in three-system VAXclusters be upgraded to at least a VAX 4000-200. This ensures that the cluster operates optimally by minimizing the disparity between adapter and processor performance. Use of the same DSSI adapter (SHAC) on two machines also guarantees that adapter performance is balanced. See Table 1–1 for differences in adapter performance. The 4000-series Ethernet adapter (SGEC) is also the highest performing Ethernet adapter.

How the configuration is defined will be based on several factors. Among the important factors are overall system performance, storage capacity, and data availability. In some cases user needs can be met with one of these factors taking precedence. In other cases, a conflict will result and will require resolution.

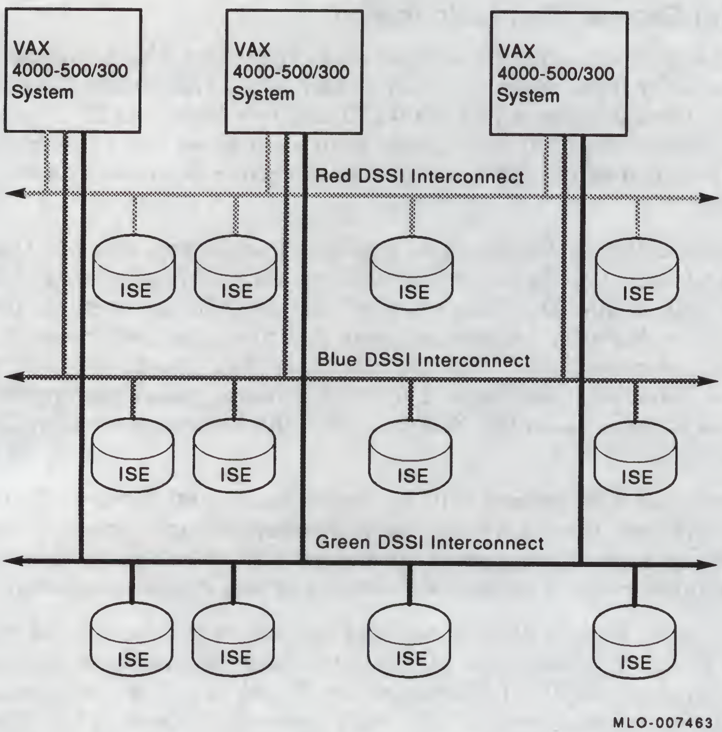
When configuring for highest VAXcluster performance, it is important to maximize I/O capacity. To do so, ensure that all cluster storage is directly accessible to all processors residing on the DSSI interconnects. Speed varies proportionately with directness of access, because the fewer devices intervening between a user's terminal and the disk on which required information is stored, the less time an I/O request spends waiting for service. (The state of waiting for service is referred to as latency.) Direct connection eliminates server overhead also.

2.6.2 Configuring for Storage Capacity

If configuring for maximum capacity is preferable, Mass-Storage Control Protocol (MSCP) can be used to serve storage to cluster members. Highest MSCP service performance is obtained by having the most powerful VAX systems with the highest-performing adapters act MSCP servers and by ensuring that those VAX systems are not overloaded with other processing tasks. Secondly, it is preferable to locate heavily used files on devices with direct storage and relegate less active files to served devices.

Note that although the cluster in Figure 2–3 provides excellent storage capacity, it runs the risk of lost access to storage as a result of a single-path access to most ISEs.

Figure 2–2: Directly Shared ISEs for High Performance



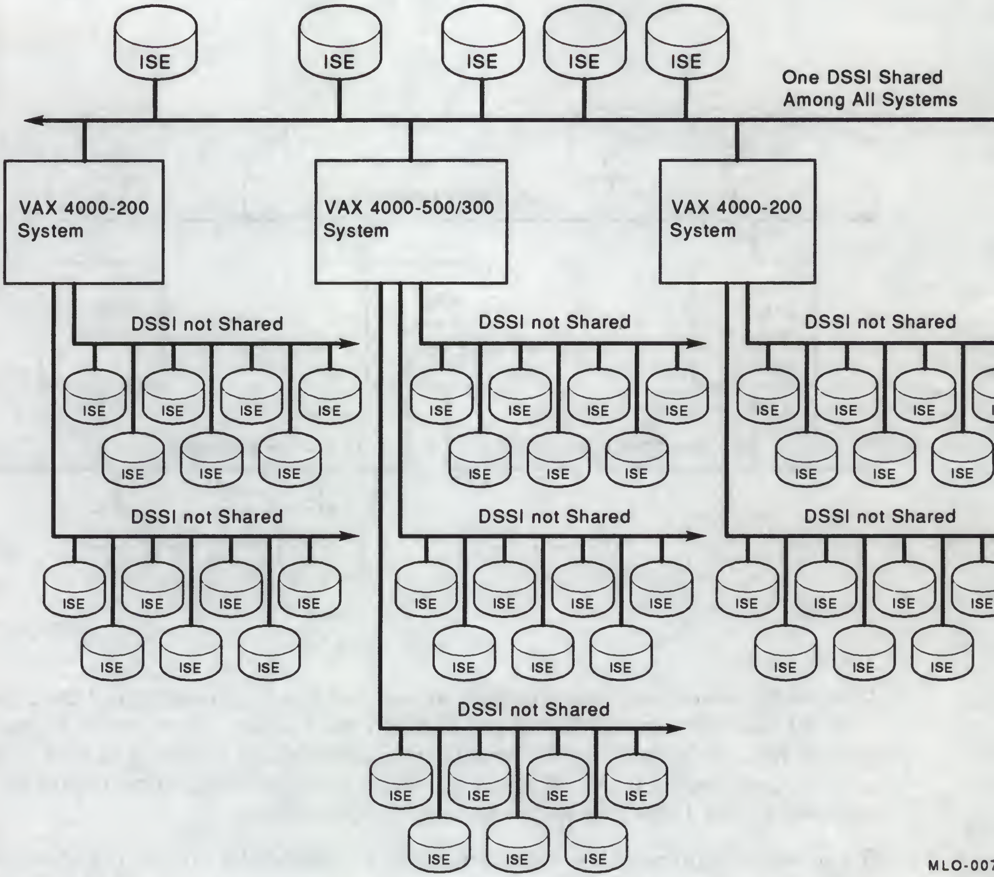
2.6.3 Configuring for Data Availability

Access to storage upon failure of a processor is a key factor in creating and maintaining maximum data availability to users, especially in configurations without direct storage. Access to data is ensured by attaching two MSCP servers (and therefore two processors) to the ISE, enabling failover to occur if one processor fails.

Another requirement for data availability is to put ISEs in different enclosures than the processors. This means that if a breakdown that necessitates a power shutdown occurs in an enclosure it will not affect both the processor and ISE.

It is wise to have at least two boot nodes, so that if one becomes unavailable satellites can boot from the other.

Figure 2–3: Configuring for Capacity

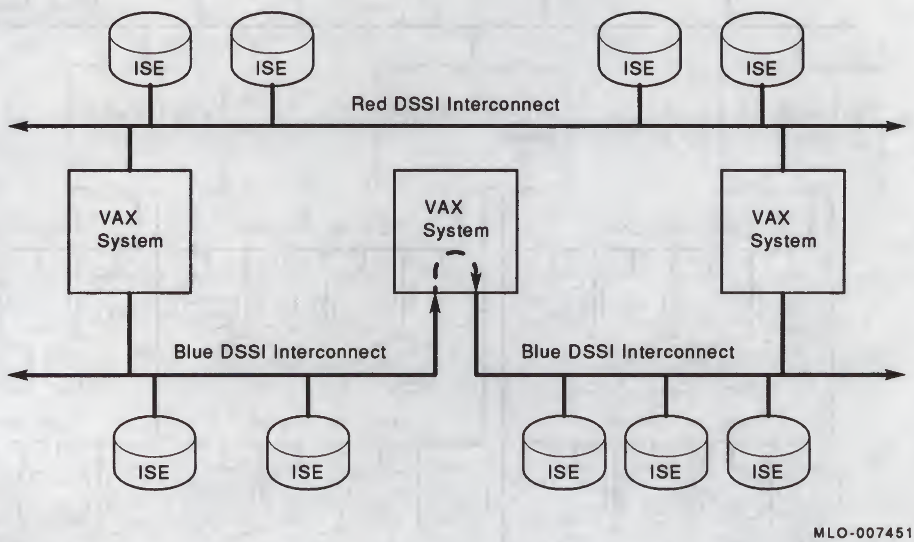


2.6.3.1 Using Multiple DSSI Interconnects for Redundancy

The DSSI Interconnect is DC coupled and uses low-level signals. DC coupled means that devices, connectors, ISEs, cables, adapters, and so on are not isolated electrically. In contrast, CI VAXclusters are AC coupled and are isolated electrically. AC coupling allows you to disconnect cables while systems are running without causing electrical or data-integrity problems. DC coupling does not. DSSI does not allow you to disconnect a cable, short circuit a connector, or in any other way affect the electrical characteristics of a pathway while the cluster is running. Therefore, anything that happens on an interconnect or pathway affects all devices connected to that interconnect.

For example, if you need to install a dual-host VAX 4000-500 configuration and you have four ISEs to attach to it, what arrangement should you use to connect them? On the VAX 4000-500s, you have two embedded DSSI adapters. You could put all the

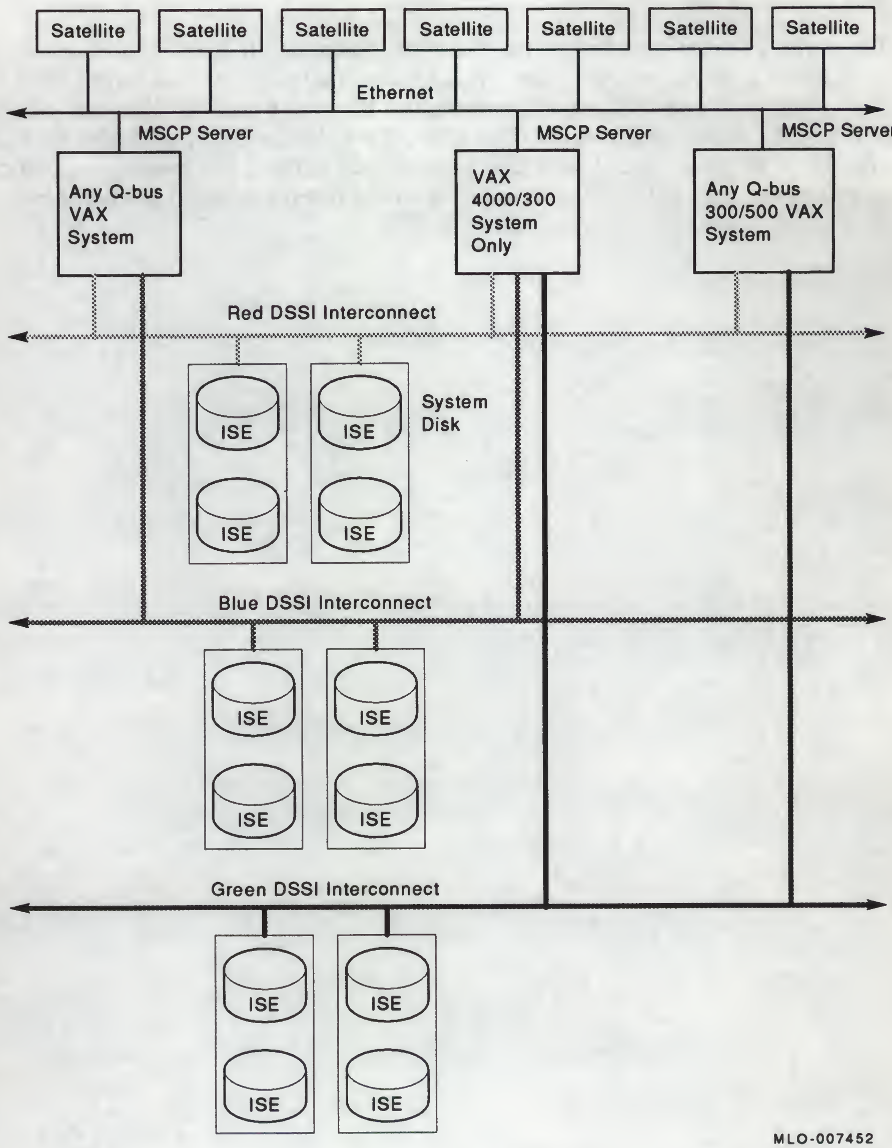
Figure 2–4: Configuring for Availability



ISEs on the first interconnect (which we will call bus 0). After you did the installation with all the ISEs on bus 0 and powered up the system, what would happen to the system if, for example, one of the bus transceiver chips on ISE 3 failed? If the fault on the chip was such that it short circuited one of the signal pathways to ground, then the entire DSSI bus would be in a fault condition.

If you were to bring the system down and put two ISEs on bus 0 and two ISEs on bus 1, the situation would not be eliminated. However, the likelihood of both buses failing in the same mode at the same time is remote. By distributing the ISEs over the two buses, you would isolate the effects of a bus failure and avoid building in a single point of failure. Avoidance of a single point of failure is one of the most basic rules for any configuration. Furthermore, use of two interconnects provides a second path for interprocessor communication in the event that one interconnect fails. With two buses, you can also split the I/O load by by placing half the ISEs on each bus. By adding Volume Shadowing, you could duplicate the data on multiple ISEs connected by different buses. Thus, loss of a bus would not affect cluster operation, since the shadow members on the remaining bus could continue to operate and provide service.

Figure 2–5: Using Multiple DSSI Interconnects for Redundancy



2.6.3.2 Using Volume Shadowing to Prevent Loss of Access or Data

Common system disks are normally used to minimize the system management of three-system VAXclusters. For that reason, it is wise to make use of Digital's volume shadowing software to maintain multiple copies of the system disk for use

if hardware fails. With volume shadowing, if a failure occurs (such as if the system disk or its power source or bus breaks down) the system can utilize another member of the shadow set.

The same principle applies to user or data disks. If all data is kept on one ISE or on ISEs on a single interconnect, upon failure the users will not have access to their data. Worse, if the ISE becomes corrupted they may lose it. You can prevent loss of access by shadowing critical data ISEs. This shadowing should also be used from one bus to another. You should also keep critical ISEs in different storage enclosures to prevent loss of data in the event of a power failure within the enclosure.

Chapter 3
Installation

This chapter outlines the installation and configuration of a DSSI VAXcluster. It provides you with a basic understanding of the procedures involved in installing and configuring a DSSI VAXcluster but is not exhaustive in its coverage of general VAXcluster topics. Detailed information is available in other manuals, such as the *VMS VAXcluster Manual*, the *VMS Upgrade and Installation Manual*, the *VMS License Management Facility Manual*, the *Guidelines for Configuration of VAXclusters* and the documentation set that accompanies your system hardware. Please read chapters 1 and 2 and this chapter before beginning the installation.

3.1 Information Required To Install a VAXcluster

Before beginning the installation, you must obtain the information that you will need to complete it. As you obtain this information, you must keep a written record, as you will need it at various points in the installation process and also in future maintenance. The information you will need is listed below, and each list item is explained in the text that follows the list:

- The cabling arrangement of the hardware devices.
- A DECnet node name, DECnet node address, and allocation class for each VAX system.
- A VAXcluster group number if the cluster will include satellites.
- A VAXcluster password if the cluster will include satellites.
- A quorum value.
- An SCS node name, DSSI node ID number, allocation class number, and unit number for each ISE.

3.1.1 DECnet Node Name and Address

The DECnet node name, which is also the SCS node name for processors, cannot exceed 6 characters. Obtain a DECnet node address for each VAX system from your network manager.

3.1.2 VAXcluster Group Number and Password

You need a VAXcluster number and password only if the cluster will include satellites. A cluster group number uniquely identifies each local-area cluster on a single Ethernet. This number must be in the ranges of 1 to 4095 or 61440 to 65535. The password provides a second level of validation to ensure the integrity of clusters on the same Ethernet that may have the same group number. It also controls access by preventing an intruder who has discovered the group number

from joining the cluster. The password cannot exceed 31 characters, consisting of the alphanumeric characters plus the dollar sign and underscore. See the *VMS SYSMAN Utility Manual* for further information.

3.1.3 Allocation Class

The allocation class is a naming convention used in VAXclusters to provide processors with path-independent access to RF and TF devices that have multiple access paths. It is a number from 1 to 255 that is used to create a device name in one of the following formats:

```
$allocation-class$device-name
```

For example, the device name 1DIA1 identifies an ISE that has multiple access paths between two VAX systems that have an allocation class of 1. The allocation class operates so that when a system not directly connected to a device attempts to access it, it can do so by means of any path. The allocation class for a system should match the allocation class for the ISEs to which it is directly connected. Setting the allocation class to 0 sets no allocation class for a device.

3.1.4 Quorum Disk

-

The principle of quorum exists to prevent destruction of data caused by a processor's inadvertent simultaneous membership in two clusters. Quorum operates so that each system or quorum disk receives a specific number of votes (set by the SYSGEN parameter VOTES), and a number determined by the following formula allows the cluster to function only if the required number of members is present:

$$QUORUM = (EXPECTEDVOTES + 2)/2$$

In this formula, EXPECTED_VOTES is the sum of all votes held by potential cluster members in the cluster plus a value for the quorum disk. In a dual-host VAXcluster, designating an ISE as a quorum disk adds a third vote, so that quorum can be reached with only one system functioning. During certain cluster state transitions, quorum is calculated somewhat differently. For further information on these differences, see the SYSGEN manual in the VMS documentation set. Note that only one disk can be a quorum disk in a single VAXcluster and that a quorum disk cannot be volume shadowed. Satellites customarily have a quorum value of 0. For more information about quorum, consult the *VMS System Generation Utility (SYSGEN) Manual* and the *VMS VAXclusters Manual.*

3.2 Installing the Hardware

This section contains a sequential list of the tasks you will need to perform to install the hardware for a DSSI VAXcluster. The items in the list are explained in the remaining sections of the chapter. The tasks are as follows:

1. Create a schematic diagram for the VAXcluster.
2. Label each VAX system and ISE with identifying information.

3. Label each hardware element (including bulkhead connectors and adapters) with one of the supplied colored stickers, according to which DSSI interconnect it will be connected to.
4. Cable together the hardware devices.
5. Configure DSSI node IDs on the processors.
6. Turn on the processors and check to ensure that the self tests complete successfully.
7. Obtain I/O addresses for ISEs served by KFQSA adapters if you plan to use them.
8. Configure the KFQSA adapters if you plan to use them.
9. Configure the ISEs.
10. Verify the status of the DSSI interconnect and ISEs on each processor.
11. Set the parameters for booting on each processor.

The following is a list of console commands for a VAX 4000-500 or 4000-300. Many are explained in this section. For information on those that are not, consult the documentation accompanying your VAX processor. Commands on other consoles may be different from the those listed here.

BOOT [[/R5:]<boot_flags>] [<boot_device>]
CONFIGURE
CONTINUE
DEPOSIT [<qualifiers>] <address> <datum> [<datum>...]
EXAMINE [<qualifiers>] [<address>]
FIND [/MEMORY | /RPB]
HALT
HELP
INITIALIZE
MOVE [<qualifiers>] <address> <address>
NEXT [<count>]
REPEAT <command>
SEARCH [<qualifiers>] <address> <pattern> [<mask>]
SET BFLG <boot_flags>
SET BOOT <boot_device>
SET CONTROLP <0..1 | DISABLED | ENABLED>
SET HALT <0..4 | DEFAULT | RESTART | REBOOT | HALT | RESTART_REBOOT>
SET HOST/DUP/DSSI/BUS:<0..1> <node_number> [<task>]
SET HOST/DUP/UQSSP </DISK | /TAPE> <controller_number> [<task>]
SET HOST/DUP/UQSSP <physical_CSR_address> [<task>]
SET HOST/MAINTENANCE/UQSSP/SERVICE <controller_number>
SET HOST/MAINTENANCE/UQSSP <physical_CSR_address>
SET LANGUAGE <1..15>
SET RECALL SET RECALL <0..1 | DISABLED | ENABLED>
SHOW BFLG
SHOW BOOT

SHOW CONTROLP
SHOW DEVICE
SHOW DSSI
SHOW ETHERNET
SHOW HALT
SHOW LANGUAGE
SHOW MEMORY [/FULL]
SHOW QBUS
SHOW RECALL
SHOW RLV12
SHOW SCSI
SHOW TRANSLATION <physical_address>
SHOW UQSSP
SHOW VERSION
START <address>
TEST [<test_code> [<parameters>]]
UNJAM

3.2.1 Labeling Processors and ISEs with Identifying Information

During the installation procedure, you will obtain much identifying information on the hardware devices. You will need this information later in order to maintain the VAXcluster. To preserve this information, Digital strongly recommends that you create a label containing this information for each VAX processer and ISE. Use Table 3–1 and Table 3–2 as templates for creating your labels. Note that some label categories include identifying information that is programmed into the hardware or software (for example, Allocation Class), and therefore are explained in this chapter, whereas others (such as Enclosure) merely serve as a means of distinguishing one piece of equipment from others of its kind in the same VAXcluster.

Table 3–1: Template for VAX System Labels

VAX System (for example, 4000-500)	______
DECnet or SCS Node Name	______
DECnet Address	______
DSSI Node ID (Adapter __)	______
DSSI Node ID (Adapter __)	______
DSSI Node ID (Adapter __)	______
DSSI Node ID (Adapter __)	______
Allocation Class	______
Unit Number	______
DSSI Interconnect	______
VAXcluster Group Number	______
VAXcluster Password	______
Quorum Disk	______
Enclosure (for example, R400Xa, R400Xb)	______

Table 3–2: Template for ISE Labels

Hardware Device (for example, RF73)	______
SCS Node Name	______
DSSI Node ID (Adapter ___, Bus __)	______
DSSI Node ID (Adapter ___, Bus __)	______
DSSI Node ID (Adapter ___, Bus __)	______
DSSI Node ID (Adapter ___, Bus __)	______
Allocation Class	______
Unit Number	______
Controller's System ID	______
HISPEED Setting (0,1)	______
Force Name Setting (0,1)	______
Q-Bus Address (KFQSA Adapters only)	______
DSSI Interconnect	______
Enclosure (for example, R400Xa, R400Xb)	______

3.2.2 Cabling the Hardware Elements Together

Configuration flexibility is a primary feature of DSSI VAXclusters. Such flexibility brings with it a myriad of configuration possibilities that include many different hardware devices. The complexity of the task of cabling devices together increases with the number of devices. For this reason, Digital strongly recommends that you create a schematic diagram of your configuration before attempting to cable the elements together.

3.2.2.1 Resolving Configuration Issues Before Cabling

A schematic diagram allows you to visualize the connections between the components and to resolve configuration issues before making physical connections. It provides sufficient information to view the configuration without the complexity of cabling and enclosures. With logical configuration issues resolved, cabling becomes much less difficult.

Use the logical representation to help make decisions about where to place ISEs, how to cable together the various DSSI enclosures, ISEs, and system adapters. Determine which interconnect is connected to which processor and ISEs, bearing in mind, for example, that if you plan to use an ISE as a shadow volume, you should not put the shadow disk in the same enclosure as the disk being shadowed or in any other position (such as on the same interconnect) where a single point of failure could make both disks unavailable simultaneously. Another factor to be aware of in volume shadowing is that the disk write function is not finished until both the active disk and the shadow disk are written. Thus, if the shadow disk is served by MSCP, it will slow down I/O performance. Performance is also affected when adapters of different performance levels serve the active disk and shadow disk.

Figure 3–1: Logical Example of a VAXcluster Configuration

TBD

3.2.2.2 Cabling the Hardware

To organize the process of cabling the entire cluster, and to make it easier to diagnose problems in the configuration later, Digital is providing (in the hardware kits) a set of colored stickers for identifying the hardware components. With the labels you can define and document each critical DSSI connection within the configuration. In addition, the labels make it possible to understand the organization of the VAXcluster when someone other than the installer must maintain or repair it. Please disregard the instructions written on the sticker package.

Figure 3–2: Cabling Example of a VAXcluster Configuration

TBD

3.2.3 Using KFQSA Adapters

This section describes the requirements of KFQSA adapters. (If you are not using these adapters, you can skip this section of the manual.) If you are planning to use a KFQSA adapter, it is necessary to program its controller with a Q–bus address for each ISE attached to it so that the DSSI interconnect and the software can communicate with the ISEs. You must perform this step before you can configure the ISEs. Unlike the KFQSA, embedded adapters do not require this type of programming.

3.2.3.1 Determining I/O Addresses for Q–bus Devices

You must provide addresses for the simulated controllers on the Q–bus (ISEs and KFQSA adapters themselves) so that the DSSI interconnect and the software can communicate with them. You can determine the Q–bus addresses for all ISEs from the console of any VAX machine except a MicroVAX II by entering the console command CONFIGURE. If all your VAX systems run on MicroVAX II machines, you must use the MicroVAX Diagnostic Monitor (MDM) to determine what the addresses should be and to set them. For information on using MDM, consult the *KFQ Storage Adapter Installation and User Manual.* (For general information on Q–bus addresses, consult the chapter of that manual entitled "Programming the Configuration Table Using Console Commands.")

The CONFIGURE command prompts you for the name and number of each Q–bus device to be configured and then generates Q–bus addresses and vectors for each device. Vectors identify devices to the software when I/O interrupts occur. The default number of devices is one.

It is important that Q–bus devices are configured correctly, as errors can result in I/O devices not being seen by the software and therefore being unavailable. Such errors are often misinterpreted as hardware malfunctions. Devices accessed via a KFQSA adapter must also be configured with Q–bus addresses and vectors. KFQSA adapters simulate storage port controllers and therefore use I/O addresses allocated to permanent and floating MSCP and TMSCP storage controllers.

Type HELP at the CONFIGURE command prompt to determine the format of devices names, and then enter your response as follows:

```
Device? Number?TK70
Device? Number?KFQSA-DISK,4
Device? Number?DEQNA
Device? Number?EXIT
```

Type EXIT when you have entered all your devices. The CONFIGURE command responds as follows:

```
-774500/260 TK70
-772150/154 KFQSA-DISK
-760334/300 KFQSA-DISK
-760340/304 KFQSA-DISK
-760350/314 KFQSA-DISK
-760354/320 DEQNA
```

The 6-digit numbers are the Q–bus I/O addresses, and the numbers following the slash are the vectors. The default number of devices is one. You must set the addresses and vectors for each Q–bus option using the appropriate mechanism on the Q–bus option. However, vectors are automatically set by the software for the KFQSA. For information on how to set addresses and vectors, consult the hardware documentation accompanying each device.

Keep a record of the order of these addresses now, as you will need them later. If you are planning to use two KFQSA adapters, you will need both the addresses you have set and the remaining available addresses when you are programming the second KFQSA adapter.

3.2.3.2 Programming KFQSA Adapters

KFQSAs are protocol converters that provide an interface between the Q–bus and DSSI. The KFQSA performs its interface function by representing each ISE it is attached to as a UNIBUS–Q–bus Storage Systems Port (UQSSP) controller (such as the RQDX3 or the KDA50) with one device attached. If you had a KFQSA with 6 ISES attached to it, for example, the KFQSA would appear on the Q–bus as 6 UQSSP controllers when you were finished programming it.

```
>>>>SHOW QBUS
Scan of Qbus I/O Space
-200000DC (760334) = 0000 (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK)
-200000DC (760340) = 0000 (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK)
-200000DC (760344) = 0000 (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK)
-200000DC (760350) = 0000 (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK)
-200000DC (760354) = 0000 (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK)
-200000DC (774500) = 0000 (300) RQDX3/KDA50/RRD50/RQC25/KFQSA-DISK)
```

You could program the addresses of these ISEs into two KFQSAs also. If you put three ISES on each KFQSA, the two KFQSAs would appear on the Q–bus exactly as the single KFQSA did.

Use the console command SHOW QBUS to determine if any addresses have been set previously. It is important that you verify this information by looking at the hardware as errors may result if you are unaware of programmed addresses.

NOTE

Removing a device that is accessed via the KFQSA without reconfiguring all the KFQSA devices significantly slows the performance of the console when you use the commands SHOW UQSSP or SHOW DEVICE. The slowed performance is caused by timeouts occurring on programmed devices that are not connected.

Accessing the KFQSA to set the addresses from the console varies according to whether the KFQSA has been programmed previously. The next two sections explain setting the addresses in each situation.

3.2.3.2.1 Accessing an Unprogrammed KFQSA

To access a KFQSA that has not been previously programmed, you must first set the hardware switch 1 on the KFQSA circuit board switchpack to the ON position (1). Setting this switch forces the KFQSA to one of the four known service addresses (0,1,2,3) determined by switches 3 and 4. When the KFQSA is in service mode, enter the following command at the console to access it:

```
>>> SET HOST/UQSSP/MAINTENANCE n
```

Replace *n* with a digit between 0 and 3, which tells the SET HOST command which service address to use. Entering this command creates a connection to the KFQSA and displays the contents of the configuration table if there are any.

To add the three KFQSA disks from the previous example, enter the following command, where the number following the SET command is the DSSI node ID number and the number following the address is 21 for disks (22 for magnetic tapes). The DSSI node ID number is a digit between 0 and 7 that allows the software to communicate with the storage devices.

```
? SET 0 772150 21
```

Enter the SHOW command to display what you have entered. To remove an entry, enter the CLEAR command with the node number or set another value. Enter EXIT when the addresses are correct.

3.2.3.3 Accessing a Previously Programmed KFQSA

If the KFQSA has already had Q–bus addresses programmed into it, you may use one of these addresses to access the adapter without having to remove the KFQSA to place it in service mode.

SHOW UQSSP displays information on all UQSSP ports accessible via the KFQSA. A port is an interface between hardware and software, and UQSSP is the protocol by used to access these ports. The output from the SHOW UQSSP command is as follows:

```
>>>>SHOW UQSSP
UQSSP Disk Controller 0 (772150)
-DUA0 (RF30)
```

In this example, 0 is the controller number for the first disk controller being simulated by the KFQSA.

Enter the following command to access the KFQSA:

```
>>>SET HOST/MAINTENANCE/UQSSP/SERVICE{/DISK/TAPE} n
```

Replace *n* with the controller number obtained from the SHOW UQSSP command. Use /DISK if the controller number is for a simulated disk controller and /TAPE if it is for a tape controller. You can now program the KFQSA as you would an unprogrammed KFQSA in Section 3.2.3.2.1. (Another method of accessing the KFQSA without putting it into service mode is to use the command SET HOST /MAINTENANCE/UQSSP and a VAX I/O address for a simulated controller. VAX I/O addresses are listed in the display from the SHOW QBUS command.)

3.2.3.4 Setting DSSI Node ID Number for a KFQSA Adapter

You also need to set a DSSI node ID number for the KFQSA itself. At the KFQSA prompt (?), enter the following command to set the DSSI node ID for the KFQSA:

```
SET n\KFQSA
```

Replace *n* with the DSSI node ID number that you want to assign to this KFQSA adapter. You have now programmed the KFQSA. You must now power down the system, remove the KFQSA circuit board, and set switch 1 to the OFF position (0) to return the KFQSA to operating mode.

3.2.4 Configuring the ISEs

Intelligent Storage Elements (ISEs) provide functions similar to those of Hierarchical Storage Controllers (HSCs). In installing ISEs, you must program them with identifying information so that VMS can communicate with them. In addition to performing disk and tape service functions, ISE operating environments provide other utility functions that can be accessed by users from the console or from VMS. Access is provided by a special protocol called Device Utility Protocol (DUP).

The console command SHOW DEVICE lists the DSSI node numbers that you will need to connect to the ISEs. The SET HOST command that you use to access the ISEs varies according to the adapters you use. Table 3–3 lists the commands according to the adapters.

Table 3–3: SET HOST Commands According to Adapter Type

System	Adapter Type	Command
VAX 4000-300 & 4000-500	Embedded	SET HOST/DUP/DSSI/BUS:[0,1] dssi_node_number port_controller_number[1] PARAMS[2]
VAX 3300, 3400, & 4000-200	Embedded	SET HOST/DUP/DSSI dssi_node_number PARAMS
All systems	KFQSA	SET HOST/DUP/UQSSP port_controller_number[1] PARAMS

[1]The port controller number obtained by using the SHOW UQSSP command

[2]To access other functions, use a different utility name. Use DIRECT with the command to get a directory of the utilities available.

To prepare the ISE for installation, you access the ISE utility task PARAMS. PARAMS controls display and modification of many different operating parameters including those that govern ISE operation with the cluster. The parameters you need to modify are the following:

- Force unit
- Node name
- Unit number
- Allocation class

Optional parameters include the following:

- Controller system ID
- HISPEED
- Force name

Use the SET command and the parameter name at the ISE prompt (PARAMS>) to set these parameters. Use the SHOW command to ensure that you have set them as you intended to, and then use the WRITE command to write these settings to the controller and thereby permanently set them.

Use the ISE command LOCATE and the device name to find the physical location of a device if you are not sure, for example, which physical device is DIA21:. This command, which is not available on the RF30 and RF71s, causes the fault light to be turned on (by soft-faulting the driver) on the named device.

3.2.4.1 SCS Node Name ISE Parameter

The SCS node name is JOE in the device name JOE$DIA21:. For purposes of simplifying and clarifying the organizational structure of DSSI VAXclusters, Digital recommends that you use the node name parameter to encode the color of the DSSI bus to which each ISE is connected. For example, the ISE JOE$DIA21: would become R_JOE$DIA21: if it were connected to the DSSI bus with the RED sticker.

If it were connected to the DSSI bus with a RED AND WHITE sticker, it would become RW_JOE$DIA21. Use G for green, Y for yellow, and B for blue.

Use the following command at the ISE prompt (PARAMS>) to set the node name parameter:

```
SET NODENAME [bus_]nodename
```

The node name cannot exceed 6 characters and may include the the underscore character.

3.2.4.2 Unit Number ISE Parameter

The unit number is 21 in the device name JOE$DIA21:. Unit numbers should not be less than 10. The first digit (2) refers to the DSSI bus number and the second (the least significant digit, in this example 1) to the ISE DSSI node ID number. In the device name JOE$DIA121, the DSSI bus number is 12, and the DSSI node ID number is 1. This numbering system allows you to see instantly which bus an ISE is connected to and which physical ISE is which DSSI node. Use the following command at the ISE prompt (PARAMS>) to set the unit number parameter.

```
SET UNITNUM n
```

3.2.4.3 Allocation Class ISE Parameter

The allocation class is explained in Section 3.1. Note that the ISE allocation class parameter is spelled *allclass* whereas in VMS the same paramter is spelled *alloclass*.

Use the following command at the ISE prompt (PARAMS>) to set the allocation class parameter.

```
SET ALLCLASS n
```

3.2.4.4 Force Unit ISE Parameter

You must also set the force unit parameter to 0. This turns off the force-unit parameter which is enabled in the factory-installed software when it arrives at the customer site, and whose function is to make the DSSI unit number equal to the DSSI node number. The unit numbers must be different if you are planning to use more than one interconnect, as having unit numbers identical to the DSSI node numbers would mean that you could have two ISEs with a unit number of number of 0. To enable the force unit number, set it to 1. If your VAXcluster will use more than one interconnect, Digital recommends that you not use the values 0 through 7 as unit numbers. By avoiding these numbers, you avoid having two devices with the same unit number in the event that you fail to set the force unit to 0.

Use the following command at the ISE prompt (PARAMS>) to set the force-unit parameter:

```
SET FORCEUNI 0
```

3.2.4.5 Optional ISE Parameters

The parameters in this section are optional, but you may wish to use them after the installation.

3.2.4.5.1 Controller's System ID

The parameter SYSTEMID is a 48-bit address automatically set on the ISE by the factory. It identifies the storage device for the software. It becomes of interest when you wish to replace an ISE and you want the new ISE to be seen by the system exactly as if it were the old ISE. You can achieve this by changing the system ID on the new ISE to the system ID on the old ISE.

3.2.4.5.2 HISPEED

The HISPEED parameter is automatically set to 0 (disabled) by the factory. When it is set to 1 (enabled), the speed of disk access is increased but at the expense of capacity; only one-half normal disk space is available for use.

3.2.4.5.3 Force Name

The FORCENAM parameter when set to 1 (enabled) disables the NODENAM parameter and forces the SCS node name for the ISE to be the name of the hardware device (such as RF72) followed by a letter identifying the DSSI bus.

3.2.5 Verifying the Status of the DSSI Interconnect and ISEs

At this point, you should verify that all the programmed ISE information is visible from all systesm. Use the console commands SHOW DSSI to verify the status of nodes on the DSSI interconnect that are accessed through embedded adapters. Use the SHOW UQSSP to display information on all disks and magnetic tapes that are accessed by means of the KFQSA. Use SHOW DEVICE to verify the status of all DSSI nodes.

3.2.5.1 ISE Naming Conventions at the Console Level

An ISE device name appears differently depending upon several factors as follows:

- Whether the adapter is embedded or not
- Whether you are using more than one adapter
- The parameters set in VMS (See Appendix C for information on VMS naming conventions.
- The parameters set in the disk drive

For example, an RF72 with a unit number of 12, an allocation class of 1, and a node name of RICKY would appear as follows:

- On VAX 4000-300

  RICKY$DUA12 if located on the first KFQSA
  RICKY$DUH12 if located on the second KFQSA
  RICKY$DIA12 if located on a SHAC A
  RICKY$DIB12 if located on SHAC B

3.2.6 Configuring Processors for Booting

In clustered configurations, the installer must ensure that each system boots the correct system disk and that it selects the appropriate system root from the system disk it is booting. It is also important to determine the action the processor will take upon a power up/power fail condition. The boot parameters control these actions and must be programmed to perform them.

In order to program the boot parameters, you must put the system into console mode. Press the HALT button on the machine or consult your hardware documentation for information on other methods of putting the system into console mode. To set the boot device, enter the SET BOOT command at the console prompt (>>>) with a device name, as follows:

```
>>> SET BOOT DIA0:
```

Do this on each processor directly connected to the system disk. Satellites boot through Ethernet. Consult your hardware documentation for information on the correct command for booting Ethernet.

To direct each processor that is directly connected to the system disk to boot from a system root, use the SET BFLAG command on each processor, as follows:

```
>>> SET BFLAG n0000000
```

Replace *n* with the number of the root directory (such as 0 for SYS0 or 1 for SYS1). These numbers are hexadecimal and must be within the ranges 0 to 9 or A to F. Boot flags are passed to VMB through hardware memory register 5 (R5). The BFLAG parameter value is automatically loaded into R5 at boot time. The number of the system root you wish to boot from is encoded in the highest nibble of the 8-digit hexadecimal BFLAG parameter.

3.3 Installing the VMS Operating System

You can install VMS either from a distribution tape or from a disk containing factory-installed software (FIS). Using FIS eliminates the need to restore the software from the distribution tape. This section describes the procedure for installing from a distribution tape. Information on installing factory-installed software is in the *VMS Factory-Installed Software Installation Manual.* For additional information on installing VMS, consult the *VMS Upgrade and Installation Manual.*

To install the operating system from a distribution tape, you first need to install Standalone BACKUP, which is part of the distribution media. The VMS software is supplied on the distribution tape in the form of save sets, which must be restored in order to be used. Standalone BACKUP allows you to perform the restoration operation.

To install Standalone BACKUP, place the appropriate tape in a drive, and enter the BOOT command from the console, as follows:

```
>>> BOOT ddcu
```

Replace *ddcu* with the name of the drive containing the magnetic tape. When Standalone BACKUP has been successfully installed, remove the magnetic tape device and replace it with the device containing the operating system.

To install VMS, enter the following command at the console:

```
>>> BACKUP/IMAGE/VERIFY source-drive:save-set.BCK/REWIND target-drive:
```

For information on installing VMS, see the *VMS Upgrade and Installation Manual.*

To boot VMS, enter the BOOT command from the console, as follows:

```
>>> BOOT ddcu
```

Replace *ddcu* with the name of the drive on which you have installed VMS.

To install licenses for the operating system, the VAXcluster, and DECnet, invoke the VMS License Management Facility as follows:

```
$ @SYS$UPDATE:VMSLICENSE
```

For information on installing licenses, consult the *VMS License Management Facility Manual.*

3.4 Installing and Configuring a VAXcluster

The steps involved in creating a VAXcluster after you have installed VMS are as follows:

- Install DECnet.

  To install DECnet, run NETCONFIG from the directory SYS$COMMON:[SYSMGR], as follows:

  ```
  $ @NETCONFIG
  ```

  The directory SYS$COMMON:[SYSMGR] contains most of the files you will need to create a VAXcluster. For most questions with DECnet and other such procedures, it is wise to choose the default answer if you are unsure of the correct answer. The information you provide to DECnet is placed in the system's permanent database. At the end of the procedure, NETCONFIG asks you if you want to start the network. Answer YES to this question, because you need to have the network running to configure the second and third systems as cluster members.

- Run CLUSTER_CONFIG.COM (@CLUSTER_CONFIG).

  CLUSTER_CONFIG.COM, which is documented in the appropriate VMS installation manual for your hardware, allows you to make decisions about how you want the cluster to run. If you do not know the answer to a question, it is wise to accept the default answer. Questions include whether you want to enable conversational bootstraps, which device you want to use for paging and swapping, and whether you want to use Ethernet for interprocessor traffic.

Conversational bootstraps are bootstraps that use a dialogue that allows you to modify VMS parameters by means of SYSBOOT and pauses to receive your answers.

Use the default for the answer to the question on the paging and swapping device, which is the system device, except for satellites that have a local disk. In such cases, using the local disk for paging and swapping cuts down on network traffic.

Regarding the use of Ethernet for cluster traffic, answer YES if you are using KFQSA adapters (DSSI does not carry interprocesser traffic if you are using KFQSA adapters) and if you are using satellites.

The VMS installation or upgrade procedure generates a master file directory that contains a pointer to a common root directory on which most operating system and optional product files are stored. This directory also contains pointers (in the form of directory aliases, such as SYS0.SYS and SYS1.SYS) to system root directories for each node in the cluster. CLUSTER_CONFIG.COM creates system root directories for you and also creates the aliases. The logical name SYS$SYSROOT is automatically defined as a search list that points to the local root first (SYS$SPECIFIC) and then to the common root (SYS$COMMON). Thus, the logical names for system directories (such as SYS$SYSTEM and SYS$LIBRARY) point to two directories, a local root (for example, SYS$SPECIFIC:[SYSEXE] and a common root SYS$COMMON:[SYSEXE]. For more information on the directory structure in clusters, consult the *VMS VAXclusters Manual*, and for more information on CLUSTER_CONFIG.COM, consult the *Guide to Setting Up a VMS System*.

CLUSTER_CONFIG.COM also runs AUTOGEN to set system parameters.

- Restart the network.

  AUTOGEN reboots the system after setting the system parameters but does not automatically start the network. Because the network must be running when you attempt to configure a second or third system, you must restart it from the directory SYS$COMMON:[SYSMGR] after AUTOGEN has finished, as follows:

  ```
  $ @SYS$MANAGER:STARTNET
  ```

- Add a second or third system.

  Run CLUSTER_CONFIG *on the first system* each time you wish to add another cluster member.

NOTE

Do not boot other systems until the configuration for all nodes is complete.

Appendix A
Cable Information

This appendix contains descriptions, lengths, and order numbers for cables used in DSSI VAXclusters.

Table A–1: Electrical Lengths of DSSI Interconnect Components[1]

Description	Connector Type	Part Number	Length
3.5-foot intracab shielded cable used in SF200, SF210, and SF100 cabinets to connect between the drive enclosures' SF72 and TF857/837	MR/MR[2]	BC21Q-3F	42 in., 3.5 ft., 1.06m
6 ft. (70 in.) intracab shielded cable used in SF200 /210 cabinets between drive enclosures and SF200 /SF210 bulkhead	MR/MR-BH[3]	BC21R-5L	70 in., 5.8ft., 1.78m
9 ft. external shielded cable;	MR/MR	BC21Q-09	108 in., 9 ft., 2.74m
9 ft. external shielded cable;	MR/PS[4]	BC22Q-09	9 ft., 108 in., 2.74m
9' external shielded cable;	PS/PS	BC21M-09	108 in., 9 ft., 2.74m
25 ft. external shielded cable	MR/MR	BC21Q-25	300in., 25 ft., 7.62m

[2]MR is a micro-ribbon style external shielded connector and mates with MR-BH only.
[3]MR-BH is a micro-ribbon style shielded connector used for bulkhead mounting and mates with MR only.
[4]PS is a pin and socket style external shielded connector and mates with PS-BH only.

Table A–2: Electrical Lengths of Embedded DSSI Interconnects In Enclosures

Enclosure	Connector Type	Internal DSSI Length
R400x through-bus mode; no internal terminator; up to 7 drives both upper and lower rows	2 external PS-BH[1]	94.5in., 7.875ft., 2.40m
R400x split bus mode 1; no internal terminator; up to 4 drives on the same bus, upper row only	2 external PS-BH	66in.,5.5ft., 1.68m
R400x split bus mode 2; no internal terminator; up to 3 drives on the same bus, lower row only	2 external PS-BH	40 in.,3.33ft., 1.02m
BA440 imbedded storage (BUS 0); has internal terminator; 4000-300 and higher	1 external PS-BH	52ft. approximately, 4.3ft., 1.32m
BA440 in/out port (BUS 0); no internal terminator; 4000-300 and higher	2 external PS-BH	20 in.,1.6 ft., 0.51m

[1]PS-BH is a pin-and-socket style shielded connector used for bulkhead mounting and mates with PS only.

Table A–2 (Cont.): Electrical Lengths of Embedded DSSI Interconnects In Enclosures

Enclosure	Connector Type	Internal DSSI Length
BA 430 imbedded storage; has internal terminator; 4000-200	1 external PS-BH	54 in., 4.5 ft., 1.37 m
BA213; has internal terminator	1 external PS-BH	45 in.,3.7 ft., 1.14 m
B213F; has internal terminator	1 external PS-BH	20 in.,1.6 ft., .51 m
BA215; has internal terminator	1 external PS-BH	30 in.,2.5 ft., 0.76 m
R215F; no internal terminator	1 external PS-BH	60 in.,5 ft., 1.52 m
R23F; no internal terminator	1 external PS-BH	39 in., approximately 3.3 ft., 1.0 m
KFQSA; connector directly attached to KFQSA (for example, BA440)	1 external PS-BH	0
SF72 or SF73 enclosure in through-bus mode; 1 to 4 drives on the same DSSI bus; no internal terminator	2 external MR-BH	168 in., 14 ft., 4.27 m
SF72 or SF73 enclosure in split bus mode; 1 or 2 drives using internal SF72 terminator	1 external MR-BH	83.5 in.,6.96 ft., 2.12 m
TF857 or TF837; no internal terminator	2 external MR-BH	10 in.,0.83 ft., 0.25 m

Cable lengths internal to the SF200/SF210/SF100 must be obtained by adding the intracab cable length to the lengths in the enclosures used (SF72 or TF857/837). Usually the SF100 has only a 3.5 foot intracab cable between the enclosures, and the SF200/SF210 has one or two 70-inch cables and possibly a 3.5 foot intracab cable. Consider the specific implementation.

For example, an SF200 with the bulkhead connected to a through-bus SF72 that is connected to a TF857 that is connected back to the bulkhead would have 70 + 167 + 42 + 10 + 70 = 359 inches (29 ft. 10 in.) internal to the SF200 cabinet.

Enclosures with no internal terminators may be used either on the ends or the middle of the interconnect. When they are used on the end, an external terminator must be used on the enclosure.

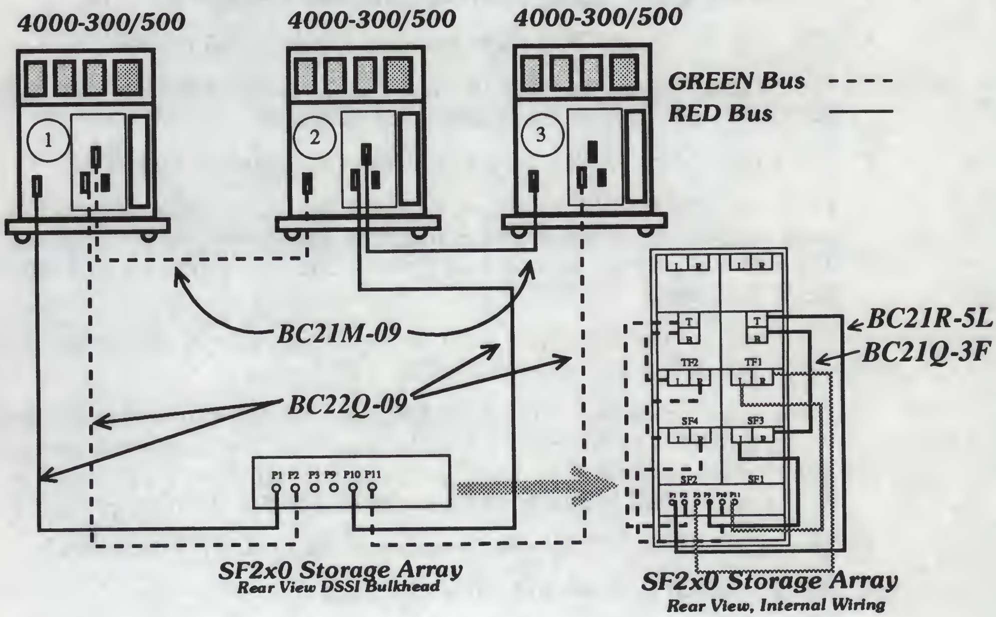
Enclosures with internal terminators can occupy bus end positions.

Appendix B
Length Information

This appendix contains information on determining lengths for the DSSI Interconnect.

To installing a DSSI VAXcluster that does not conform to the 5-enclosure rule, it is necessary to validate each DSSI bus for qualified length separately. This is accomplished by totaling up all the individual DSSI bus segments. These bus segments include all intercabinet cables, all cabling used within each enclosure (for example, cabinet bulkhead cables) and DSSI cabling that may be used within a DSSI option such as the SF72. The discussion here refers to the interconnects in the following figure.

3 processor 4000-300/500 / 2 Shared Bus DSSI VAXcluster
Physical View using 1 SF2x0 Storage Array



In order to validate the accompanying configuration the RED bus total length and the GREEN bus total lengths must be calculated.

Calculation for the RED Bus

The RED DSSI bus begins with Bus 0 in System 1. The internal wiring for the two DSSI buses for the 4000-300/500s must be considered in the total DSSI bus length calculation.

- The length of Bus 0 internally on the 4000-300/500 is 52 inches.

  The RED DSSI bus then exits System Enclosure 1 and then connects to the SF200 storage array using a BC22Q-09.

- The length of the BC22Q-09 intercabinet DSSI cable is 108 inches.

  From the SF200 bulkhead it connects to a BC21R-5L, which connects the DSSI to one of two TF857s within the SF200. You must include the DSSI segment within the TF857 as part of the total bus calculation.

- The length of the BC21R-5L intracabinet DSSI cable is 70 inches.
- The length of the TF857 Tape Cartridge Subsystem is 10 inches.

  The TF857 tape cartridge subsystem is then connected to one of the SF72 storage building blocks using a shorter intracabinet DSSI cable, BC21Q-3F. As with the TF857, you must include the DSSI segment contained within the SF72 storage building block.

- The length of the BC21Q-3F intracabinet DSSI cable is 42 inches.
- The length of the SF72 Storage Building Block is 83.5 inches.

  From the SF72, another BC21R-5L intracabinet DSSI cable is used to bring the RED DSSI bus back to the SF200 DSSI bulkhead.

- The length of the BC21R-5L intracabinet DSSI cable is 70 inches.

  From the SF200 DSSI bulkhead, a second BC22Q-09 cable (M/R to P/S) cable is used to bring the RED DSSI bus into Bus 1 of the second 4000-300/500 system. The internal cabling for Bus 1 on System 2 must be incorporated into the total DSSI bus length.

- The length of the BC22Q-09 intercabinet DSSI cable is 108 inches. The length of the Bus 1 Internal on 4000-300/500 is 20 inches.

  From bus 1 on System 2 the RED DSSI bus connects to system 3's Bus 0 using another BC22M-09 intercabinet cable. The final bus segment to consider in the total bus calculation is that utilized for the internal wiring of Bus 0 on System 3. The RED DSSI bus terminates at the end of Bus 0 within System 3.

- The length of the BC22M-09 intercabinet DSSI cable is 108 inches.
- Bus 0 Internal on 4000-300/500 is 52 inches.

  This represents all the DSSI bus segments used to implement the RED DSSI bus from terminator to terminator. Adding all these segments together should reveal whether this bus is supportable within a computer room (must not be greater than 82 feet) or within an office area (must not be greater than 65.6 feet).

- The total length for this bus is 723.5 inches (60.29 ft) which is within the limits for office or computer room.

  You must perform this exercise for the GREEN bus by following the same set of steps to determine all DSSI bus segments used to implement the total bus from terminator to terminator.

Appendix C

VMS Naming Conventions for ISEs Connected to a KFQSA Controller

In VMS Version 5.3, the device names assigned to ISEs attached to a KFQSA adapter changed. In previous versions of VMS, the device name had the form DIcu, where c, the controller letter, was A, B, C, and so forth. The controller letter was taken from the device name of the port (PUAO, PUBO, PUCO, and so forth) through which the ISE is accessed. If the allocation class n of the DSSI disk was a nonzero digit, then the device name had the format nDIcu. This scheme was inconsistent with the naming used for ISEs attached to an integrated (SII-based) adapter, such as the EDA640, which is used on MicroVAX 3300 and 3400 systems.

With the new naming scheme introduced in Version 5.3, the device name no longer depends on the device name of the port. Instead, all ISEs use the controller letter A. Thus, device names now have the format nDIAu, where n is the nonzero allocation class of the ISE, or nodename$DIAu if the allocation class is zero.

In order to alleviate some of the problems anticipated with this type of a change, the new naming scheme was dependent on a new SYSGEN parameter VMS5. With VMS5 set to 1, the old (prior to Version 5.3) device naming continued to be used, while setting VMS5 to 0 enabled the new scheme. Systems installing Version 5.3 for the first time had VMS5 set to 0 by default, while systems upgraded from a previous version of VMS had VMS5 set to 1. For example, a single KFQSA with three DSSI disk drives attached would have the following device names for ports/disks:

	Port	Disk	Port	Disk
	Allocation Class = O		Allocation Class = 4	
Old scheme: (VMS5=1)	PUAO	DIAO	PUAO	4DIAO
	PUBO	DIB1	PUBO	4DIB1
	PUCO	DIC2	PUCO	4DIC2
New scheme: (VMS5=O)	PUAO	FRED$DIAO	PUAO	4DIAO
	PUBO	Barney$DIA1	PUBO	4DIA1
	PUCO	WILMA$DIA2	PUCO	4DIA2

In VMS Version 5.4, the old naming scheme became unavailable and was replaced by the new scheme. A benefit of the new device naming scheme is that two systems in a Dual-Host configuration will always use the same device name for a shared ISE. With the old device naming scheme, which included the port controller letter for KFQSA-connected devices, a Dual-Host configuration with multiple KFQSAs per

system could have inconsistent device names across the two systems if the common DSSI was incorrectly attached (for example, if KFQSA 1 on MicroVAX A is attached to KFQSA 2 on MicroVAX B). The old scheme also precluded dual-hosting with mixed adapter types (integral SII and KFQSA).

With the new scheme, all systems (regardless of adapter type) use device names \$n\$DIAu or nodename\$DIAu. The only variable is the allocation class or node name.

Glossary

Interconnect
Hardware used by VAXclusters to access storage and to faciliate communication between individual nodes.

Bus
See **Interconnect**.

ISE
An integrated storage element is a combination of disk and MSCP server.

DSSI ID number
A number between 0 and 7 that identifies a device to which DSSI transports information and for which DSSI therefore requires an address, including ISEs and the adapters on VAX systems.

Index

T

U

V

X

digital